REGRESSING DEXTEROUS FINGER FLEXIONS USING MACHINE LEARNING AND MULTI-CHANNEL SINGLE ELEMENT ULTRASOUND TRANSDUCERS

A Thesis
Presented to
The Academic Faculty

By

Lamtharn Hantrakul

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Music Technology in the
School of Music - College of Design

Georgia Institute of Technology

May 2018

Copyright © Lamtharn Hantrakul 2018

REGRESSING DEXTEROUS FINGER FLEXIONS USING MACHINE LEARNING AND MULTI-CHANNEL SINGLE ELEMENT ULTRASOUND TRANSDUCERS

Approved by:

Dr. Gil Weinberg
College of Design
Georgia Institute of Technology

Dr. Alexander Lerch
College of Design
Georgia Institute of Technology

Dr. Byron Boots
College of Computing
Georgia Institute of Technology

Dr. Mason Bretan
Research Engineer
Samsung Research America

Date Approved: April 25, 2018

To my parents

ACKNOWLEDGEMENTS

Thank you,

to Gil Weinberg, my advisor, for giving me the freedom, encouragement, resources and trust to pursue my wildest dreams in Artificial Intelligence, Robotics and Music here at Georgia Tech and beyond

to my dear friend Zachary Kondak, for the late night sessions fixing hardware, debugging code, collecting data, chats over Reese's shakes at 3AM on the 2nd leftmost table at The Cookout, jam sessions with and without Shimon the robot and for our friendship over the last two years (with many more to come!)

to Alexander Lerch and Byron Boots, for their insightful and sharp advice in both my academic work and long-term life goals

to Mason Bretan, for being an inspiration and role-model for all things music and machine learning since my first day at GTCMT

to my undergraduate mentors, Kathryn Alexander, Konrad Kaczmarek, Roman Kuc, Thomas Duffy and Larry Wilen, for nurturing and guiding my passion and integration of science, technology, engineering and music

to my high school mentors, David Larking, Brian Taylor and Narongrit Dhamabutra for planting the seeds that have blossomed into everything I do today

to my mother, Sukanya Hantrakul, and father, Kavi Chongkittavorn, for their love and support to follow my heart and passions unhindered

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Human Machine Interfaces or HMI's come in many shapes and sizes. The mouse and keyboard is a typical and familiar HMI for interfacing with personal computers. In applications such as Virtual Reality or Music performance, an HMI that can track finger movement with high precision is often required. Readers maybe familiar with vision-based devices like the Leap Motion or hardware-based devices like Data Gloves for use in these contexts. Similar HMI's can also be interfaced with prosthetics, although the predominant sensing technology is electromyography (EMG). In the task of detecting finger movements, each of these sensors face inherent shortcomings such as limited line of sight (vision-based sensors), hindrance to regular motion (data gloves) and susceptibility to sweat and muscle fatigue (EMG). Ultrasound, a safe and non-invasive imaging technique, is viable alternative HMI interface that directly addresses each of these disadvantages.

This thesis develops a novel system enabling real-time regression of individual and simultaneous finger flexions using a machine learning system and multi-channel single element ultrasound transducers mounted on a user's wrist. To our knowledge, this is the first implementation of a system that is capable of a.) continuous, individual and simultaneous finger regressions b.) robust to arm rotations c.) achieving these criteria using single element ultrasound transducers, as low as five, instead of a full ultrasound imaging array consisting of 128 or more transducers. In the domain of HMI's for applications such as music, VR/AR and prosthetics, this technology is an integrated, tracker-less and non-computer vision alternative for detecting finger configurations. As part of this thesis, a comprehensive dataset of ultrasound signals is collected from a study of 10 participants using our custom hardware. A series of machine learning experiments using this dataset demonstrate promising results supporting the use of single element ultrasound transducers as a HMI device.

CHAPTER 1

INTRODUCTION AND BACKGROUND

The term ultrasound is used to refer to sound pressure waves over the human threshold of hearing or 20 KHz [1]. To the general public, the technology is well known as an imaging tool to detect anatomical landmarks in a developing fetus [2]. Ultrasound imaging (called US imaging from now on) is routinely employed in hospitals to non-invasively visualize anatomical structures in the human body. Thanks to the ability for ultrasound waves to travel through soft tissue [3], the technique excels at providing both sub-millimeter resolution and imaging depth [4], enough to determine areas of skin cancer [5] and resolve tumor segmentation [6]. Most importantly, US imaging has no known side-effects [7], hence its widespread use in the medical domain.

Outside traditional medical applications, several recent studies have successfully demonstrated US imaging as the basis of a human-machine interface (HMI) [1, 7, 8, 9, 10]. Although HMIs come in many shapes, sizes and sensor types – from touch-screens to voice activated systems – this thesis is concerned with HMIs used to detect anatomical changes e.g. a person flexing their fingers or moving their arms. In this space, the reader is perhaps familiar with several systems. These can be divided into three main groups: vision-based sensors, gloves and surface Electromyography (sEMG). The Microsoft Kinect is a vision

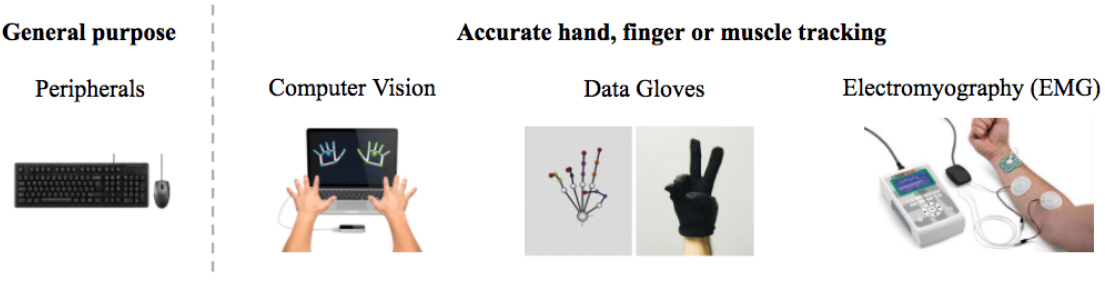


Figure 1.1: Examples of Human Machine Interfaces (HMI)

sensor that uses depth cameras to accurately detect joint angles across the human body [11]. In the entertainment industry, suits and gloves with special markers that are tracked by sophisticated camera arrays are routinely employed for computer graphics and animation [12]. Full body Inertial Measurement Unit or "IMU" suits also exist, enabling tracking of body movements using accelerometers and gyroscopes distributed across the user[1]. In the domain of sEMG, The Myo Band is a commercial example of a myoelectric sensor used to detect electrical muscle activations from the skin surface[2].

Some of these aforementioned examples are "full body" systems that may or may not have originally been designed to detect accurate finger flexions. The ability to detect individual finger movement is a crucial feature of a responsive HMI, particularly for use in music, AR/VR and prosthetic applications. In this domain, vision-based sensors like the Leap Motion use IR sensors to accurately detect finger, joint and palm position [3]. To a certain degree, the Microsoft *Kinect 2* can resolve individual finger positions, but not to the same resolution as the Leap Motion sensor. In the wearable domain, IMU gloves equipped with bend sensors and accelerometers can accurately detect finger flexions. In the domain of arm prosthetics, sEMG has seen many exciting developments, though the ability to resolve individual finger flexions using this technology is still a challenge [13]. Some of these technologies are depicted in Figure 1.1.

US imaging promises many exciting advantages over these systems, especially in the context of a HMI. Vision-based sensors like the Kinect and Leap Motion force the user to strictly remain within the system's field of view and are not intended to be moved once installed. Wearable systems like the IMU glove are often lightweight and portable, but are cumbersome to calibrate and most importantly, obstruct natural hand movement due to the presence of a semi-rigid glove equipped with resistive sensors. Interfaces designed around sEMG, though less prevalent in consumer technology, are standard in prosthetic devices.

[1]Perception Neuron Suit. `https://neuronmocap.com/`
[2]Myo Band. `https://www.myo.com/`
[3]Leap Motion. `https://www.leapmotion.com/`

	Classification (discrete)	Regression (continuous)
Ultrasound Images	Ortenzi et al. [9] Sikdar et al. (2014) [10]	Shi et al. (2010) [7] Castellini et al. (2012) [14] Gonzalez et al. (2013) [1]
Raw Ultrasound Echoes	Li et al. (2016) [15] Hettiarachchi et al. [16]	No known implementation

Table 1.1: Previous work, categorized by input type (ultrasound echoes or images) and task type (classification or regression)

sEMG is known to be susceptible to sensor crosstalk, surface sweat and muscle fatigue [9]. Implantable myoelectric systems can overcome some of sEMG's shortcomings, at the cost of being the most invasive system [10].

One of the first uses of US imaging as an HMI interface was reported by Shi et al. [7]. Features obtained through a search mechanism across the US image were used to classify finger positions. Castellini et al. and Gonzalez et al. later applied a faster and effective implementation using linear and ridge regression on hand-crafted features extracted from the US image to individually regress finger flexion [14] and finger forces [1], respectively. Sikdar et al. leveraged a hand-held mechanically scanned probe for finger action classification [10]. These studies are categorized in Table 1.1 on 3, supporting the viability of US imaging as a real-time and accurate HMI interface for detecting finger configurations.

Our previous work in US imaging for application in musical prosthetics suggest similar findings. Machine learning techniques such as Support Vector Machines (SVM) can reliably classify different finger depressions with enough accuracy and separability to enable an amputee to play piano with finger-by-finger level control [4]. Recent work by our research group suggest it is possible to use advancements in Deep Learning to regress continuous finger flexions from US images and predict arm orientations, although this is a topic for a future publication. Other successful applications of machine learning on US images include nerve segmentation [17] and a recently developed handheld US imaging device called

[4] "Skywalker" Prosthetic Hand Uses Ultrasound for Finger-Level Control. https://spectrum.ieee.org/the-human-os/biomedical/devices/skywalker-prosthetic-hand-uses-ultrasound-sensors-for-fingerlevel-control

"Butterfly IQ" [5].

The novelty of the work presented in this thesis lies in the use of a minimal array of Single Element Ultrasound Transducers (abbreviated SEUS transducers from now on) instead of a full US imaging array to perform regression of finger flexions. All aforementioned approaches employ a US imaging array, which typically contain 128 or more ultrasound transducers. The image is constructed using principles of wave reflection, whereby the strength of the returning echo is proportional to tissue and muscle density [18]. The system employed in this thesis is restricted to only 5 transducers. Imaging is thus not possible, instead, our system learns to regress finger flexions directly from raw ultrasound signals. Unlike computer vision approaches adopted previous work, the techniques employed here resemble digital signal processing (DSP) and machine learning for audio and speech recognition. By using a minimal array of SEUS transducers, our hardware and software system can be miniaturized into a portable watch-like band.

A couple of studies document the use of SEUS transducers instead of a full imaging array, notably Li et al. [15] and Hettiarachchi et al. [16]. Hettiarachchi's implementation was an entirely self contained and portable hardware system, confirming the feasibility of miniaturizing systems based on SEUS transducers into a wearable package. However, all these systems are limited to discrete classification of six hand and finger states [15, 16]. To our knowledge, this thesis is the first work that addresses the more challenging task of regressing finger flexions using a SEUS transducer system.

1.1 Contributions

We make the following novel contributions through a system that is able to:

- Regress individual and simultaneous finger flexions using a 5-channel array of SEUS transducers
- Robustly maintain and regress accurate finger flexions in different arm orientations

[5] Butterfly IQ. `https://www.butterflynetwork.com/`

- Generalize to unseen sensor locations and unseen users
- Output prediction vectors in real-time up to 70 frames per second

In addition, we collect an unprecedented dataset of synchronized ultrasound images and echoes over 10 users. The dataset contains approximately 5 Million echo samples and 850,000 US images labeled with continuous finger flexions. It is used to validate claims made in this thesis and motivate future work in the field.

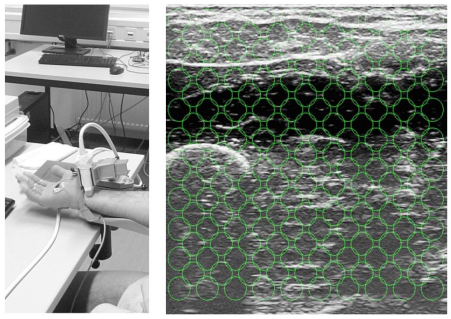
CHAPTER 2
RELATED WORK

There are largely two main threads of related work: first, publications predicting finger configurations on ultrasound images and second, predicting finger configurations on raw ultrasound echoes coupled with the the corresponding specialized hardware.

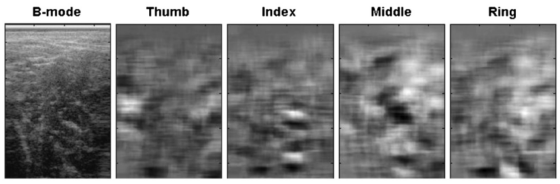
2.1 Predicting finger flexions using ultrasound images

In the domain of predicting finger flexions from ultrasound images, one of the first published implementations was achieved by Shi et al [7] in 2010. The approach first required manually identifying bounding boxes depicting muscle thickness in the ultrasound image, followed by a search strategy across each frame to maintain the location and size of these bounding boxes. The difference in bounding box size is compared to a reference value and used to control a 1 DOF prosthetic hand.

More recent approaches include work by Sikdar et al. [10] and a series of related publications: Gonzalez et al. [1], Castellini et al. [14, 8] and Ortenzi et al. [9]. Sikdar et al [10] used a K-Nearest Neighbour algorithm to classify six discrete finger classes with 98% accuracy. The authors used a hand-held mechanically scanned probe placed on the ventral (palm) side of the user's wrist. During training, US images are collected for each of the unique finger states. A mean image is computed from all samples and subtracted from each image representing each class to produce a "difference" image. Since these difference images were labeled during training, they now constitute the representative examples of a class. During inference, the mean image is subtracted from the incoming sample and the class is determined by the classes of the 5 nearest samples.



Regions of Interest or ROI
(Castellini et al.)



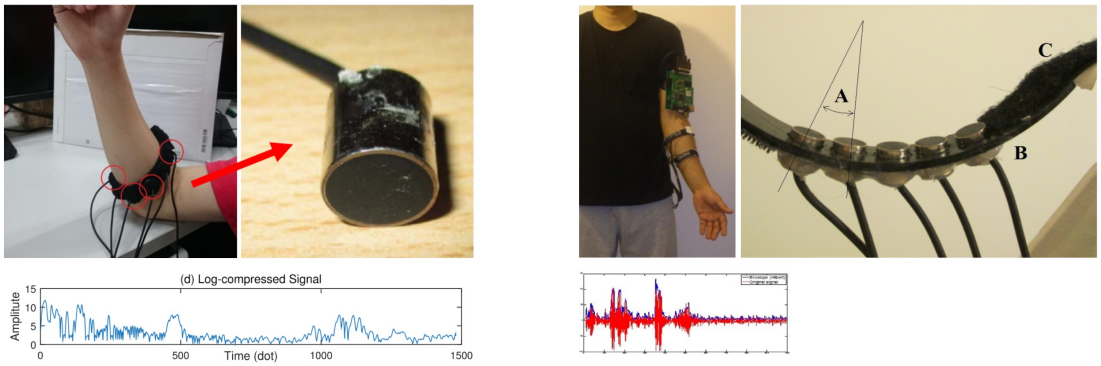
Representative "difference" images for each class
(Sikdar et al.)

Figure 2.1: **Left:** Regions of Interest (ROI) as implemented by Castelinni et al. **Right:** Characteristic difference images from each class by Sikdar et al.

2.2 Predicting finger flexions using SEUS transducers and ultrasound echoes

Gonzalez and Castelinni were able to regress individual finger flexions and finger forces respectively, in two separate publications [1, 14], by using features known as "region of interests" or ROI's on a study of 10 users. These ROI's, depicted in Figure 2.1, consist of 181 uniformly distributed circular regions of radius 20 pixels on the US image. From each ROI, three values are extracted: an offset value and two image gradients corresponding to the vertical and horizontal directions. The authors argue that these ROI's provide a compact representation of the US image and preferable over other temporal derivative image features such as optical flow. Linear and Ridge regression are then used to train a system to predict continuous finger flexions from these ROI features with nRMSE errors below 5%. These papers make two important contributions. Firstly, the regression approach works with both ground truth data being finger flexion as recorded by a data glove, and finger force, as recorded by a series of force sensors. Secondly, the system is able to incrementally update and learn on-the-fly, blurring the lines between a "training" and "inference" stage.

Ortenzi et al. [9] performed a comparative study of image features such as the aforementioned ROI's and Histogram of Oriented Gradients (HOG). They compared the performance of these features in a classification task involving 10 pre-determined gestures. From a se-



4 single element transducers
(Li et al.)

10 single element transducers in an embedded system
(Hettiarachchi et al.)

Figure 2.2: Single element transducer systems and preprocessed ultrasound echo patterns

lection of classifiers like Naive Bayes, Decision Trees and a Linear Discriminant Analysis, they concluded the best approach was a combination of a Naive Bayes classifier with HOG feature, achieving over 90% in the classification task.

In this thesis, we demonstrate regression of individual finger flexions using raw ultrasound echoes collected from a Single Element Ultrasound (SEUS) transducer array. This is a significant departure from these publications, since it circumvents imaging completely. In the related domain of SEUS approaches, Li et al. [15] developed a system consisting of 4 single elements operating at 10MHz placed near the elbow joints shown in Figure 2.2 on page 8. The authors motivated placement in these areas since they provide the transducers with full view of tendons such as the Flexor Carpi Ulnaris. The authors apply standard ultrasound preprocessing techniques such as band-passing the signal, extracting the amplitude envelope and compressing the signal in a log scale. This process is depicted in Figure 2.3 on page 9. An LDA classifier trained across the first 35 Principal Component Analysis (PCA) projections could classify six discrete finger states with an accuracy of 95%. However, the authors note that the system is very susceptible to wrist rotations and movement of the sensor. This observation is consistent with our own experimentation with SEUS transducers and is a topic of discussion later.

Lastly, Hettiarachchi et al [16] implemented an embedded and portable system con-

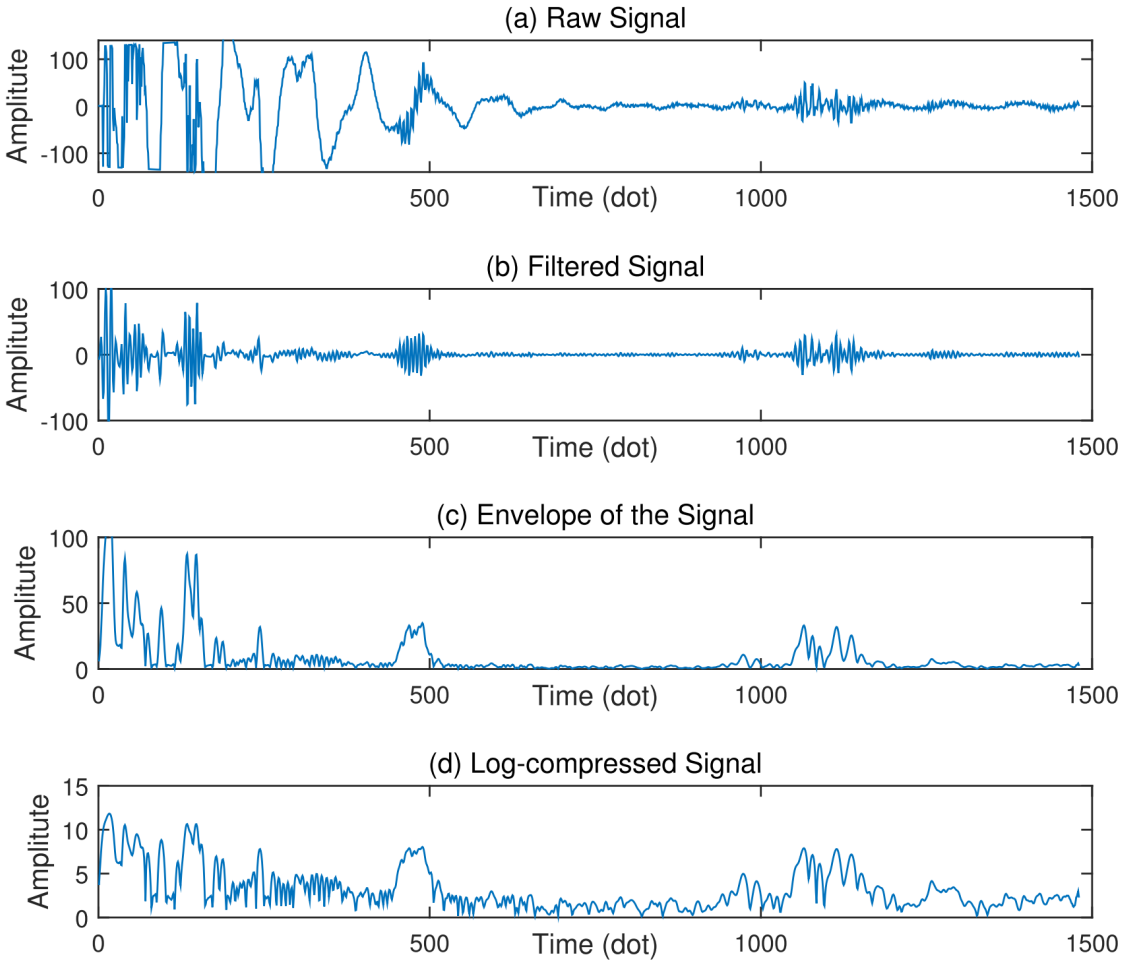


Figure 2.3: Signal preprocessing steps on the raw ultrasound echoes as implemented by Li et al.

sisting of 8 single element transducers. The authors perform similar preprocessing, band-passing the signal and taking an amplitude envelope. The paper acknowledges their main contributions to be in hardware implementation and did not perform any explicit classification. The authors showed that the signals obtained from a "training" sitting and "testing" sitting were sufficiently correlated for each finger. For example, the signals associated with an index flexion, would have low cross correlation with signals associated with the middle finger and only show strong correlation with the matching finger.

CHAPTER 3
RESEARCH QUESTIONS AND MOTIVATION

3.1 Research Questions

1. Can a function $f(x, \theta)$ be learned that accurately maps an input ultrasound echo to a continuous, real-valued output vector corresponding to finger flexions?

2. Can the function $f(x, \theta)$ be robust to arm rotations, small shifts in sensor locations and operate in real-time?

3.2 Challenges with single element data

Moving from the dense input of ultrasound image to a relatively sparse input of ultrasound echoes presents several challenges.

3.2.1 Non-imaged data

Although it is tempting to interpret the 5 single elements as "5 columns of pixels in an ultrasound image", this is not true. Ultrasound imaging probes use an array of 128 - 256 transducers in conjunction with the principal of wave reflection and linear sensor arrays to form an image. The known relationship between intra-transducer delay t, angle θ and location x assuming a plane wave is exploited to provide clear imaging across all pixels in the final image (Figure 3.1). A similar delay would need to be applied to the data from the 5 single elements in order to produce an image. Thus the ultrasound signals described in this thesis are raw ultrasound echoes as received by each of the transducers.

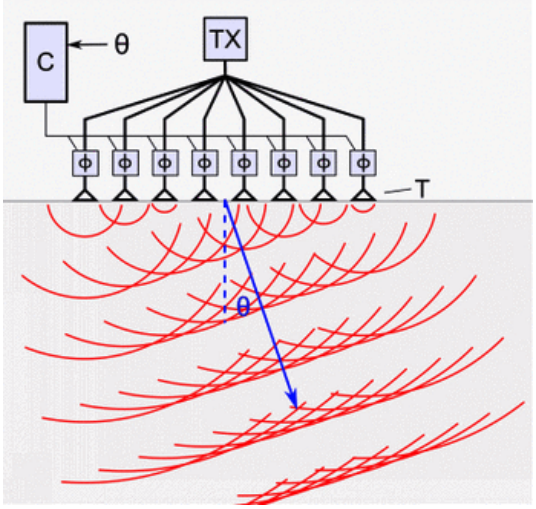


Figure 3.1: Traditional US imaging exploits the properties of a phased array with a high number of transducers. With a limited number in this thesis, imaging in this manner is not possible. (Image from `https://en.wikipedia.org/wiki/Phased_array_ultrasonics`)

.

3.2.2 Wrist and Arm Rotation

The configuration of arm muscle and tendons, as detected by both a US imaging system and SEUS transducer system, result not only from finger flexions, but also wrist flexion and rotation, as well as arm rotation and orientation. During preliminary testing with models trained on the single element data, we found signals associated with arm movements to be easily confused with signals associated with finger flexions. Hettiarachchi et al. [16] and Li et al. [15] also comment on similar problems when using SEUS transducers.

In the case of ultrasound images, one can visibly see a solid line in the ultrasound image, corresponding to the relatively thick layer of skin in contact with the imaging probe. When the user rotates their arms, the imaging probe pushes against the the users arm. This deformation appears as a shifting line that rotates in sync with arm rotations, much like a horizon line. Moreover, the relative size and spatial relationships of large scale tendon and muscle features in the image remain constant throughout these rotations. This is not the case with data from single element transducers. The characteristic pulses associated with

each element, move during both finger flexions and arm rotations, with no clear delineation between the two scenarios.

In all previous work using both US images and SEUS transducers, the user's arm and wrist must remain stationary during training and inference stages. Although our experimental hardware restricts wrist flexion and wrist rotation, it allows free arm rotation. Thus in this thesis, we address the problem of performing individual and simultaneous finger flexions with robustness against different arm orientations.

3.2.3 Summary

This thesis attempts to address the following challenges:

1. **Regression:** no prior work has demonstrated regression using single element ultrasound transducers for finger flexions

2. **Robust to arm rotations:** the models regression accuracy must remain consistent across arm orientations

3. **Robust to sensor shifts:** our approach must be able to generalize across small variations in sensor placement representative of actual use

4. **Real time:** the method should enable regression to run in real time

CHAPTER 4
EXPERIMENTAL HARDWARE AND DATASET

4.1 A Note on Authorship

This thesis was written during a cross-disciplinary collaboration between three research groups at the Georgia Institute of Technology, led by principal investigators Dr. Gil Weinberg (Robotic Musicianship Group, College Design), Dr. Levent Degerteken (Micro and Nano Engineering Lab, School of Mechanical Engineering) and Minoru Shinohara (Human Neuromuscular Physiology Lab, School of Biological Sciences). The author led the data collection presented here. With particular regards to the experimental hardware, a team of graduate research assistants, doctoral and post-doctoral students were involved in the design, fabrication and implementation of systems documented in this thesis. Unless explicitly noted in this chapter, attribute ownership of work to the thesis author. This is true for remaining chapters of the thesis.

4.2 The Ultrasound Echo-Image Dataset

The Ultrasound Echo-Image Dataset collected as part of this thesis is a comprehensive ultrasound dataset consisting of 5 Million ultrasound echo data points and 850,000 ultrasound image frames collected from 10 participants. Both ultrasound echoes and images are fully synchronized in time, collected from near-identical wrist locations and labeled with continuous values. The next sections describe the data collection protocol and important details regarding the dataset.

4.2.1 Data Collection Protocol

Our dataset is collected from a set of 10 subjects, age 25.1 ± 5 years consisting of 2 women and 8 men. Of these, 9 are right-handed, 1 left-handed and all able-bodied. Before the experiment, each subject received a thorough oral and written description of the experiment and signed an informed consent form. The experiment involving the acquisition of ultrasound signals from participants was approved by the Institutional Review Board (IRB) of our institution.

Each user follows a set sequence of actions that involves flexing the target finger as shown in Figure 4.1 on page 16. This action is repeated over a wide range of arm orientations as shown on Figure 4.2 on page 17. To ensure that signals from a finger flexion are not confused with signals from arm movement, the user pauses after flexing the target finger, using this time to move to the next arm orientation. To ensure we have enough intermediate values between open hand to a full flexion, a complete flexion motion (from open to closed and then to open) is executed over the range of 2 - 5 seconds. This is to be contrasted with data collection methods employed in Sikdar et al. [10] or Li et al. [15], in which the user rapidly transitions between different finger states with no intermediate finger flexions.

Each session takes approximately 40 minutes to complete and was collected on a new day, over a period of two weeks. Users did not report fatigue or discomfort, since the session was divided into short bursts lasting either 1 or 6 minutes each. Users were able to pause and break during each of these bursts until they were ready again.

Although this thesis is only concerned with data from single elements, the experimental hardware – explained in detail in the next section – was designed to enable data from both the imaging probe and single elements to be collected simultaneously. Although the single elements and imaging probe surface are technically not exactly on the same location, they are less than 70mm from each other. From an anatomical perspective, there is no significant difference in the tendon and muscle configurations. This would, among other future experiments, allow a comprehensive comparison of ultrasound image versus single

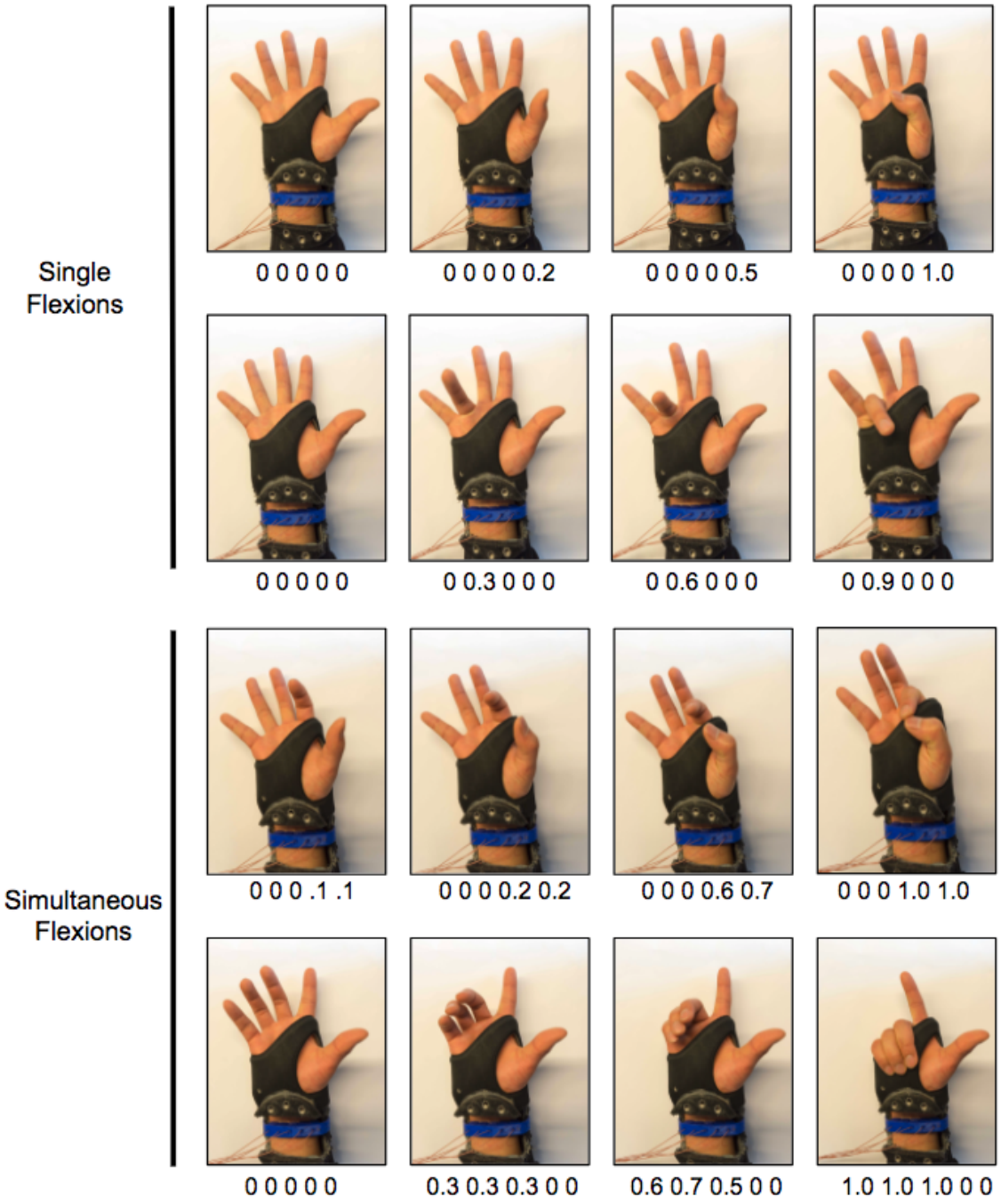


Figure 4.1: A selection of example single finger flexions and simultaneous finger flexions. Numbers correspond to the 5-dimensional ground truth flexion vector label (values truncated to 1 decimal place for visual clarity)

Figure 4.2: A sample of the different arm orientations used during data collection

element data for regression and classification tasks.

4.2.2 Dataset Structure

The structure of the dataset is summarized diagrammatically in 4.3 on page 18. For each of the 10 subjects, we collected 5 sessions of data. Each session is divided into a short and long sitting. To facilitate testing and evaluation, each finger is collected as a separate set within each sitting, labeled “thumb”, “index”, “middle”, “ring” and “pinky”. In addition, we include a “mixed” set where users perform mixed gestures consisting of simultaneous finger flexions i.e. more than one finger is being flexed at the same time and the model must regress all the simultaneous finger movements. Previous work in regressing finger flexions [1, 14, 7] do not attempt any regression beyond individual fingers.

The nature of our setup and data acquisition encourages the model to become robust against small shifts in sensor placement. The user takes off and puts back on the brace

Figure 4.3: Diagrammatic representation of the Ultrasound Echo-Image Dataset collected in this thesis. The dataset contains approximately 5 Million ultrasound echo datapoints and 850,000 ultrasound images

and sensor band between each of the 10 sittings. Thus, the dataset consists of signals acquired from 10 slightly different sensor locations. For example, model generalization can be comprehensively tested by training on 9 locations and testing on the unseen 10th location, simulating a scenario where a user has just put on the sensor band. We motivate various ways of partitioning the data in relation to different experimentation protocols in the next chapter.

4.3 Experimental Hardware Description

Our hardware for data acquisition consists of three main components:

1. a brace fitted with 5 single element transducers imaging sensor and an accelerometer
2. a medical-grade ultrasound machine used for pulsing and receiving signals from the single element transducers
3. a set of physical sliders for labeling ground truth data

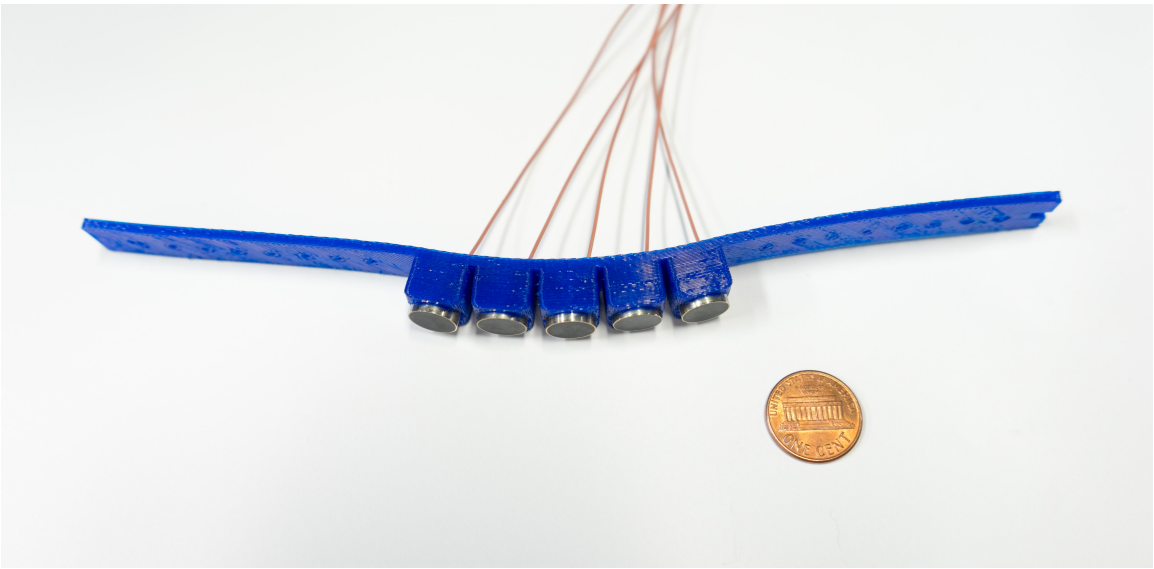


Figure 4.4: Our Single Element Transducer element housed in a custom 3D-printed band. US 1 cent coin included for size comparison.

4.3.1 Brace and onboard sensors

Single Element Transducer and Sensor Band

Each transducer is 0.55cm in diameter and 0.70cm in height; much larger than the ones found in a typical imaging array (Figure 4.5 on page 20). We hypothesizes that for the task of regressing finger flexions, it may be unnecessary to use an image. Whereas the smaller imaging sensors cast a wide beam per transducer, the larger transducer employed in this thesis cast a narrow, laser-like beam of ultrasound pulses. This is depicted in figure 4.5 on page 20. A single transducer functions both as the emitter of the pulse and the receiver of the reflected echo. The purpose of the ultrasound machine, explained later, is to control this process at extremely high sampling rates (40MHz). The user applies a small amount of ultrasound gel between the transducer surface and skin to aid conduction of these signals.

The number of employed sensors as well as location of these sensors all contribute to the accuracy of the system. In this thesis, we restrict ourselves to a small linear array of 5 SEUS transducers to resemble the form factor of a wristband. We locate our sensors near the users wrist, specifically the ventral side of the forearm. This choice of placement near the

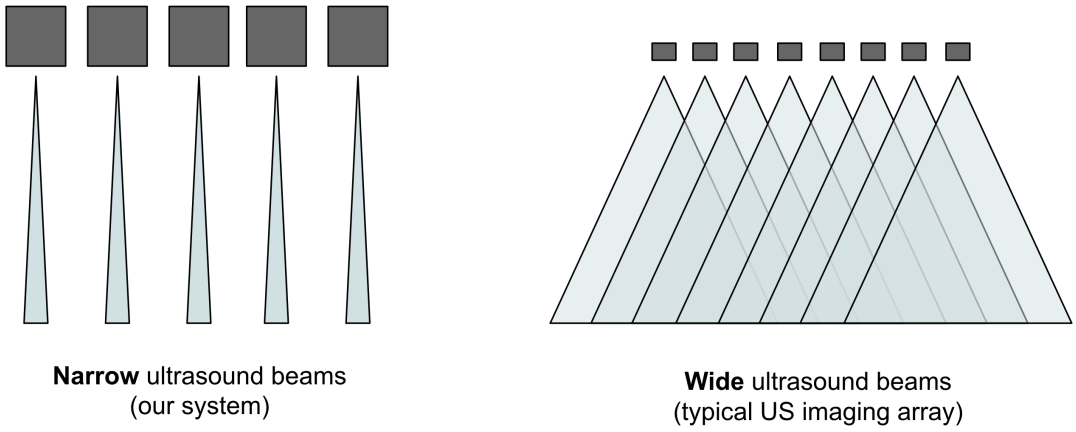


Figure 4.5: **Left:** Narrow ultrasound beams cast by our single element ultrasound transducers. **Right:** Wide ultrasound beams cast by a typical ultrasound transducer array used for imaging

wrist differs from previous work using SEUS transducers placed nearer the elbow joint [15, 16]. At the time of writing, there is no evidence supporting regression when the sensors are placed near the elbow joint, only classification. Our choice of placement matches choices from Castellini et al. [8] – who was able to perform regression at this location – and was also recommended by the principal investigator specializing in Applied Physiology. The band is also worn much like a watch and in this location the sensor has vision of the ulnar and main flexor muscles and tendons as shown in Figure 4.6. A comprehensive exploration of different sensor locations is beyond the scope of this thesis and is outlined as a future experiment.

The single elements are held in a band that was 3D printed using NinjaFlex, a flexible and bio-compatible plastic. The strap of the band contains holes like a watch, which can be secured to the brace.

Direct contact with the SEUS transducers, brace, band and gel is safe on human skin. The hardware, materials and procedures were developed by the principal investigator specializing in ultrasound transducer technology, and were approved by the Institutional Review Board (IRB) of our institution. Safety regarding the power emitted by the individual trans-

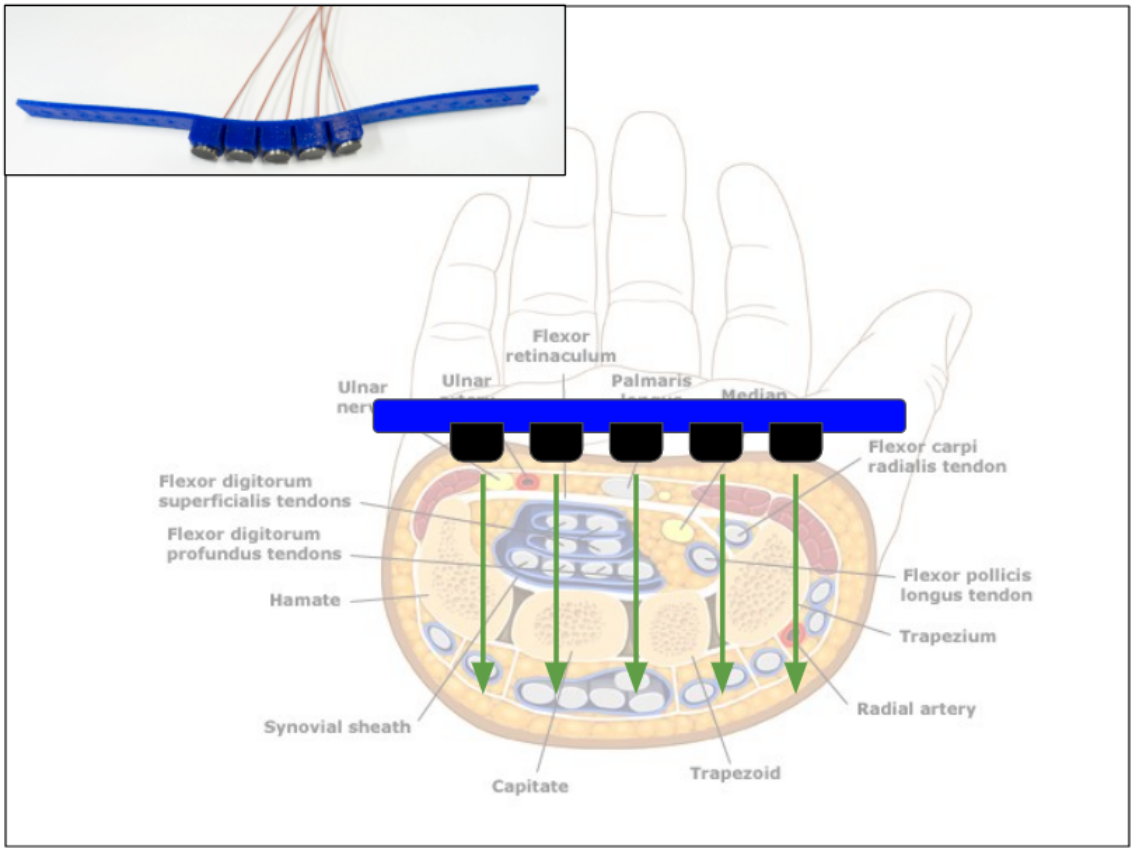


Figure 4.6: Schematic of SEUS transducer band and wrist cross-section

ducers during this process is discussed later in context of the ultrasound machine.

Authorship: The SEUS transducers were sourced by Post Doctoral Fellows Bernie Shih and Coskun Tekes. The 3D printed band was modeled and printed by Post Doctoral Fellow Chris Fink. The IRB process was written and submitted by Graduate Research Assistant Zachary Kondak.

Semi-rigid brace

Preliminary results from performing regression on both ultrasound images and raw echoes began to work reliably when the sensor was placed in roughly the same location during data collection and inference using the 1st generation brace. We ensured this by securing the imaging sensor to a semi-rigid brace as shown in Figure 4.7 on 23. The brace is oriented and locked in place via the hole where the thumb is inserted, ensuring the brace is located in approximately the same site between sittings. In the 2nd generation brace, we cut a large, additional hole to provide space for the imaging sensor and single element band.

To secure the imaging probe, we 3D printed a custom male and female receiver that firmly secures the probe to the brace, thus pressing the imaging sensor at the end of the probe tightly against the users wrist as shown in 4.7 on 23.

The 1st generation brace used a watch buckle, while the 2nd generation brace developed by the author replaces this mechanism with small screws protruding on either end of the brace. When the band is strapped onto both sides, it is held down by elastic tension. This 2nd design enables the user to quickly put on and take off the single element onto different locations, a useful feature during both data collection and prototyping different locations.

We note how it is possible for the users to flex their fingers towards the brace and then depress against the surface of the brace to induce further tightening of the tendons. In this dataset, the users were instructed to only lightly rest their flexed finger on the surface of the brace; this point denotes maximum flexion.

Lastly, as noted previously, semi-rigid brace prevents wrist rotation and flexion, but does

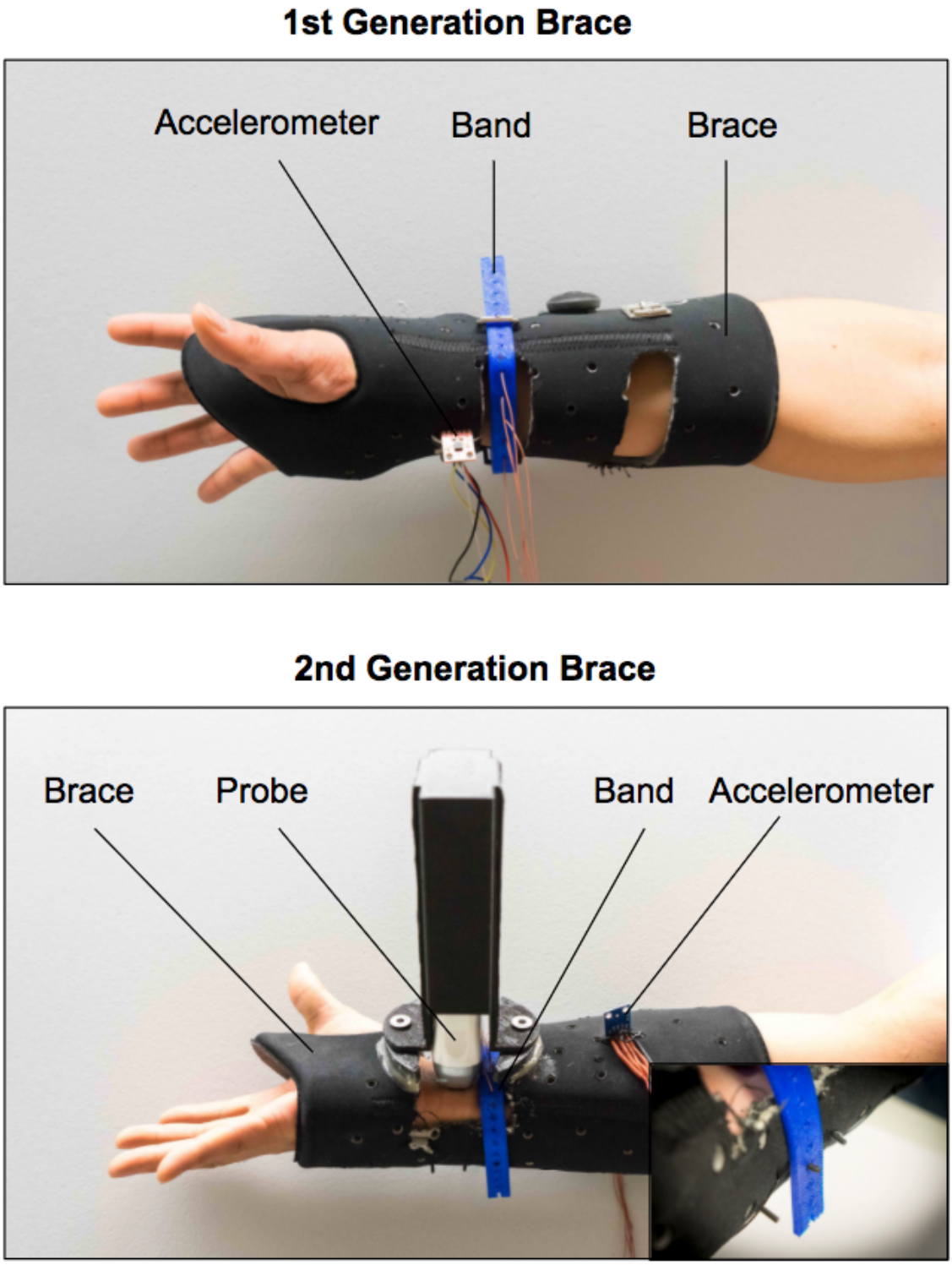


Figure 4.7: **Top:** 1st Generation brace with SEUS transducer band and accelerometer. The thumb hole ensures the sensor is placed in roughly the same location during each sitting. **Bottom:** 2nd Generation brace with both SEUS transducer band and imaging probe. Insets show improved attachment mechanism and space for an imaging probe

allow the arm to move freely to different orientations.

Authorship: The 1st generation brace and specialized 3D printed male and female adapters were designed by Post Doctoral Fellow Chris Fink. Slight modifications were added by Graduate Research Assistant Keshav Bimbraw. The 2nd generation brace was designed and implemented by the author.

Accelerometer

To collect orientation of the arm, we use a 3-axis accelerometer strapped to the brace and connected to an Arduino micro-controller. The Arduino is set with serial communication at a baud rate of 57600, well above the global data collection rate of 70Hz. The accelerometer measures the "tilt" of the brace and reports X,Y,Z values in the range 0-180. These values move in relation to the brace as the user positions their arm in different orientations as shown earlier in Figure 4.2 on 17. Although this is not as detailed as recording full joint angles and positions along the arms, each arm orientation of interest can be uniquely identified by these three values and is considered sufficient for the task at hand.

4.3.2 Ultrasound Machine

We use a medical-grade ultrasound machine *Ultrasonix SonixTOUCH* that can pulse and receive signals from both standard medical imaging probes and non-standard single element ultrasound transducers such as the one used in this thesis. Each of the single elements are connected to a single channel on the *Ultrasonix* and pulsed sequentially at a sampling rate of 40MHz. The single element transducers are rated with a bandwidth between 2MHz - 8MHz, centered on 5MHz (comprehensive frequency response chart is included in Appendix A.1.

According to the The US Food and Drug Administration (FDA), ultrasound has no bio-effects if the Mechanical Index (MI) is kept at 0.1-0.3, well below the maximum FDA limit of 1.9 [19]. In our case, the MI is calculated based on peak rarefaction or negative pressure. Even with 5 sensors at the highest possible power setting on the *Ultrasonix* machine, the

MI is measured at a safe 0.02. At these levels, there will be no memory in tissue associated with ultrasound exposure. A plot of pressure and MI provided at different power settings is included in the Appendix A.2)

Figure 4.8 on page 26 shows the pulses received from the each of the single element transducers. We explain the signal topology in greater detail in section 4.5 alongside relevant preprocessing techniques.

Authorship: The testing and calibration of SEUS hardware, alongside the configuration of the *Ultrasonix SonixTOUCH* and code for streaming frames of data via UDP, were implemented by Post Doctoral Fellow Bernie Shih and Coskun Tekes.

4.3.3 Ground Truth Annotations

Many different types of sensors can be used to collect ground truth values corresponding to continuous finger flexions. As this was an important consideration in the formation of this dataset, we survey the advantages and disadvantages of each system.

Vision-based sensors

Advantages: Systems like the Leap Motion use both specialized IR sensors and a physical model of the hand to output precise finger and joint locations. A study in 2013 [20] puts the accuracy of the Leap Motion at 0.7 mm.

Disadvantages: In our tests, we note that the Leap Motion has blind spots associated with hand positions that occlude some or all of the fingers. For example, the Leap Motion cannot accurately detect finger flexions when the hand is held perpendicular to the sensor (like a karate chop motion). Given the high sampling rate of our system, a short lapse of data provided from the Leap Motion can greatly affect the ground truth labels. Moreover, our data collection protocol involves participants moving and twisting their arm to different locations that fall outside the range of a Leap Motion.

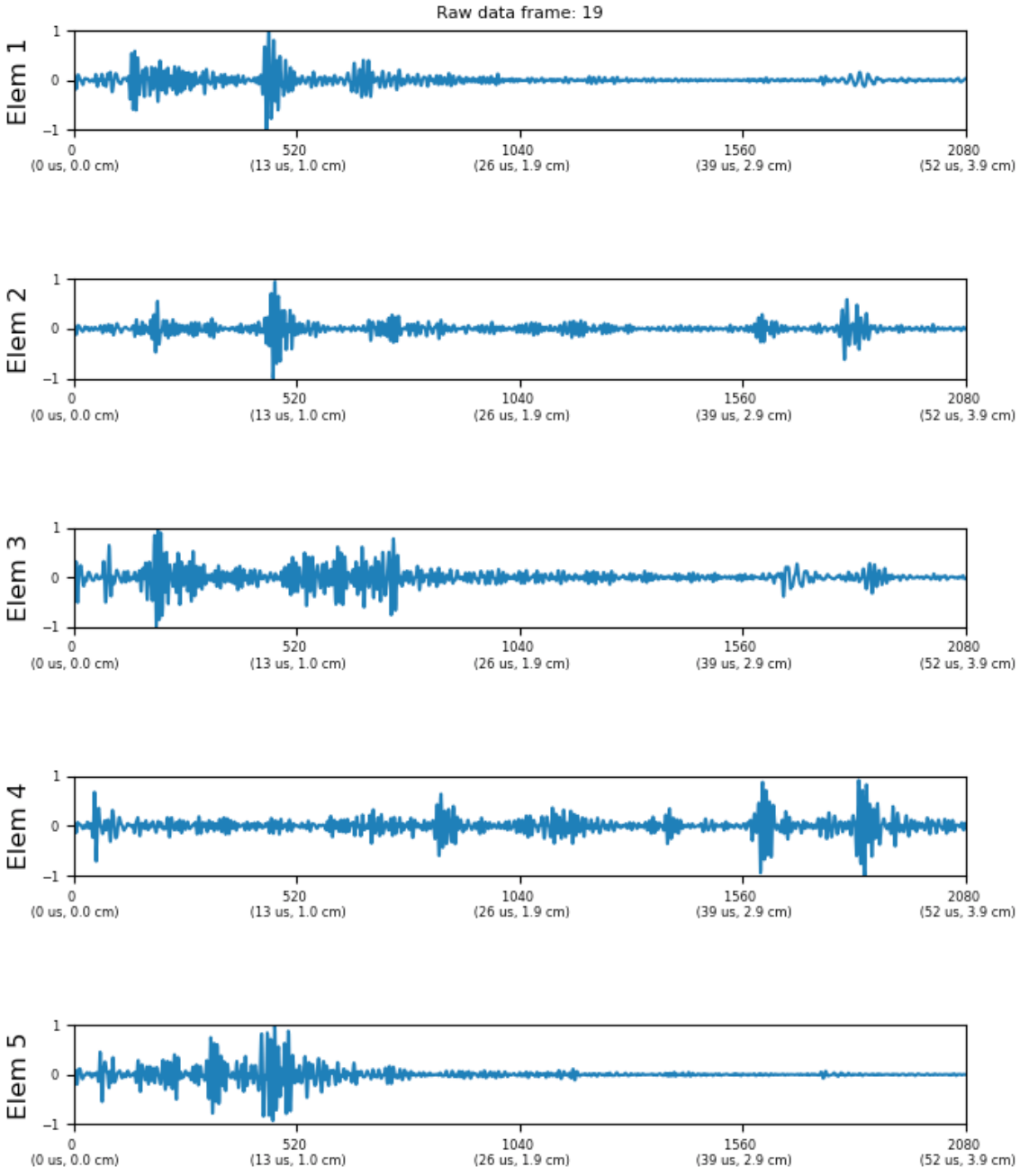


Figure 4.8: Ultrasound echo patterns received from the 5 SEUS transducers

Data Gloves

Advantages: A glove equipped with 5 bend or hall-effect sensors is arguably one of the most robust and accurate methods. Castellini et al [14], as well as many other research projects involving hand tracking, use this method as a standard approach. Unlike vision based sensors, data gloves are not limited by sensor field of view. However data gloves can be cumbersome and difficult to calibrate.

Disadvantages: Cumbersome, difficult to calibrate.

Pressure/Force Sensors

Advantages: Although a force sensor does not measure finger flexion directly, the force exerted from a finger is directly correlated with tendon activations in the wrist. Castelini et al. [8] have previously shown linear relationships between the force exerted by a finger and visual features extracted from a US image.

Disadvantages: Our experimental protocol involves moving the arm to different orientations. We would require a mechanical system that fixes the force sensors to the arm as the user moves around.

Synchronizing to an on-screen animation

Advantages: A user can watch and synchronize their finger movements to an animation of a 3D hand model. Since the sequence of finger movements in the animation is known, the ground truth labels can be synchronized with the input ultrasound data. Castellini et al. [8] as well as implementations by our group have successfully used this approach for both regression and classification using US images.

Disadvantages: We note how both these experiments collected data over short time periods. In preliminary testing, we found it was difficult for the user to remain focused and attentive to the animation at all times when the period of data collection exceeds approximately 20 minutes. This causes inaccuracies in the collected data over long periods of

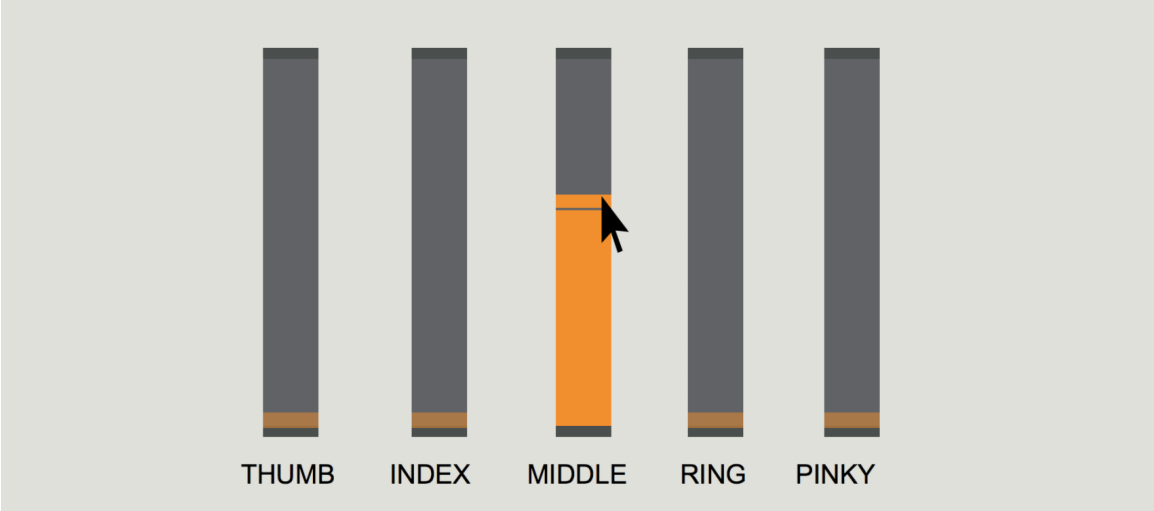


Figure 4.9: GUI interface containing a series virtual sliders. The user moves the corresponding finger slider using their free hand to indicate the degree of flexion on their flexing hand

time.

Virtual sliders

Advantages: We previously employed a simple set of virtual sliders in a GUI that correspond to the five fingers to be flexed (Figure 4.9 on page 28. When a user flexes a finger, they also use their opposite free hand to move a computer mouse and move the corresponding virtual slider to the correct value. We successfully used this method to collect values for regression using ultrasound images. We found that this approach was easier for users over longer periods of data collection, since the user employs their own proprioception to synchronize the movements and if a short break is required, the user can momentarily hold the action and label, before moving on to the next flexion.

Disadvantages: It is impossible to flex a finger without moving other corresponding fingers. This is particularly true for middle, ring and pinky fingers. For example, the tendons to the middle and ring fingers are tied, meaning a flexion of one finger invariably moves the other. When the user provides the label using a slider, only the intended finger is registered and labelled, whilst all other fingers are ignored. Unfortunately, it is too difficult of a task to

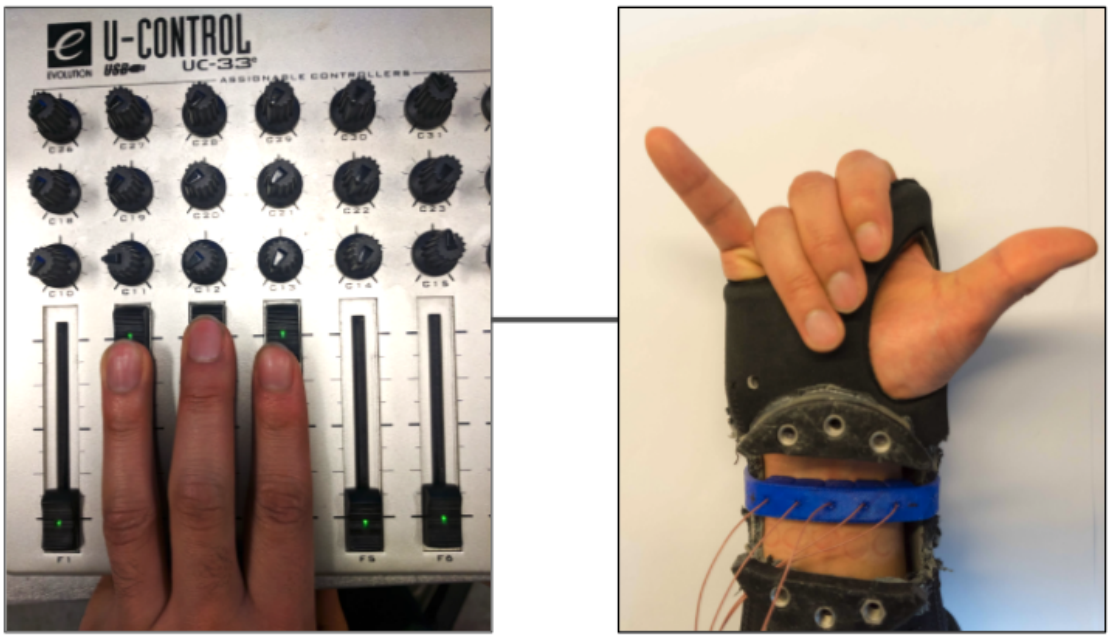


Figure 4.10: Physical sliders containing a series of linear potentiometers. The user mirrors the flexions of each finger using their opposite, free hand

require users to also estimate this involuntary finger movement in addition to the voluntary one. Data collected in this manner thus has an inherent degree of inaccuracy.

Final Approach

We decided to build upon the virtual sliders by using a corresponding set of five large physical sliders as shown in Figure 4.10 on page 29. Although the slider method is relatively more inaccurate compared to other approaches as discussed above; the main reason for doing so is a practical one. If the data collected to train the models is acquired from sophisticated and specialized sensors, it will force the end-user to have the *same* sophisticated hardware during calibration by the end-consumer.

In this regard, it is more practical to design the experimental approach around the type of data we expect the end-user to be able to provide. At the time of writing, over 2.1 billion

people own a smartphone [1]. In the context of this thesis, we make the reasonable assumption that the end-user owns or has access to a smartphone device. In this light, the physical sliders can be easily ported into a smartphone application that the user holds in their opposite, free hand, eliminating the need for any specialized hardware. Moreover, the use of physical sliders instead of a mouse and GUI enables simultaneous flexions to be labeled. In context of a final smartphone application, multi-touch can be used instead of separate physical sliders to enable users to label simultaneous flexions.

4.4 Data frame-rates

4.4.1 Ultrasound echo sampling rate

Although the *Ultrasonix* machine uses a sampling rate of 40MHz to operate beyond the Nyquist frequency of the ultrasound range, this sampling rate is a unrelated value to the rate at which the *Ultrasonix* is pinged for a reading. An external clock source, supplied by a hardware signal generator, determines how many readings per second the *Ultrasonix* machine should make. In this experiment, we collect ultrasound echo data at a rate of 70 Hz. The *Ultrasonix* machine then sends this data via UDP over Ethernet to a master desktop computer.

4.4.2 Synchronizing frame rates

Unfortunately, the Analog-Digital Converter (ADC) on the *Ultrasonix* machine cannot be configured to pulse both a standard imaging probe and our custom single element transducers at the same time. To collect both ultrasound echo and image data simultaneously, we use the *Ultrasonix* machine connected to the single element band for ultrasound echo data, and a commercially available handheld ultrasound imaging probe sold by *Sonostar* to obtain the ultrasound images.

[1] Number of smartphone users worldwide from 2014 to 2020. urlhttps://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/

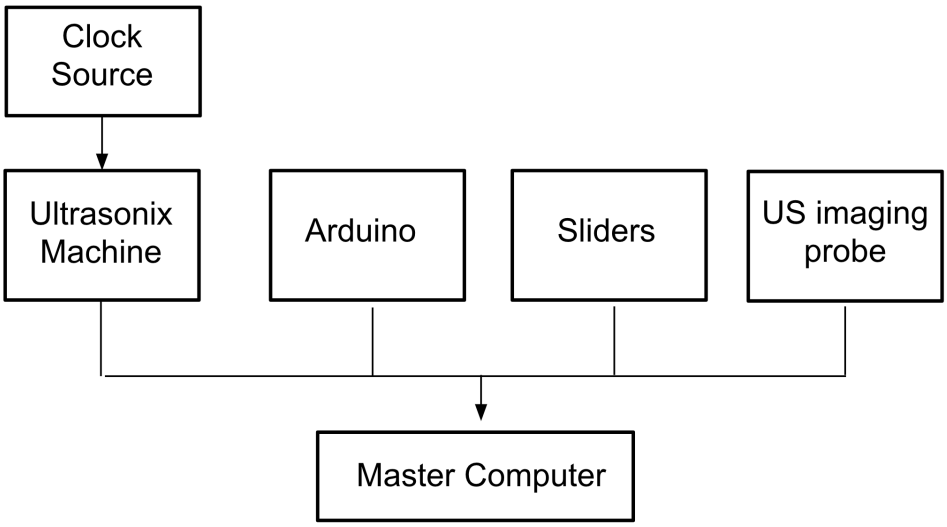


Figure 4.11: Data collection system overview. The global rate of data collection is controlled by the external clock source set at a rate of 70Hz.

Unlike the *Ultrasonix* machine, small handheld and portable imaging probes like the *Sonostar* do not have APIs with easy access to the obtained images. To circumvent this problem, we screenshot the images obtained from the *Sonostar* on a desktop computer using a screen recording application. The output from this process is a movie file and a text file corresponding to the ground truth labels for every frame in the movie. The *Sonostar* produces images on average at 15 fps, much lower than the *Ultrasonix*. To synchronize data from both sources, both the ultrasound images and ultrasound echoes are time stamped using a central maintained by the master computer. These are summarized diagrammatically in Figure 4.10.

Authorship: Graduate Research Assistant Zachary Kondak and the author contributed equally to the hardware and software data collection pipeline.

4.5 Topology of a frame of data

As previously mentioned, we collect ultrasound echoes at a rate of 70 Hz. Each frame of data is a 5x2080 matrix; the first dimension of 5 corresponds to the five single elements. Each

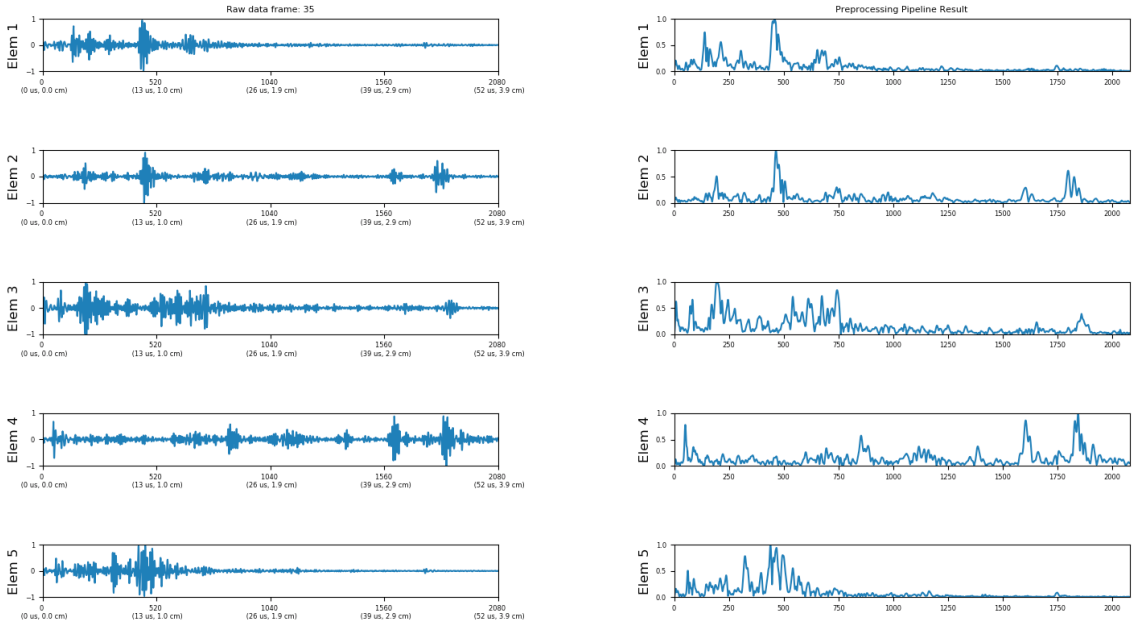


Figure 4.12: **Left:** Raw ultrasound echo patterns. **Right:** Preprocessed ultrasound echo patterns

division in the second dimension of 2080 corresponds to 25 nanoseconds. This represents a total time frame of 5.2 microseconds across the y axis. The pulses are the received signals at each transducer after an impulse is fired into the arm. Assuming the speed of ultrasound pulsed by the *Ultrasonix* machine travels at 1500 m/s [4] in human tissue, this amount of time corresponds to 3.9 cm. Thus a single frame of data encapsulates approximately 4 cm of tissue "depth" and the y axis should be thought of as distance rather than time. Each 5x2080 has a raw 16 bit range of values.

We emphasize the presence of two very different time scales. The first time scale refers to the extremely short intervals between ticks associated with sampling at ultrasound (the dimension corresponding to value 2080). As discussed before, these should be thought of as "distance". The second, and relatively longer time scale, refers to the rate at which the ultrasound machine is pinged for a reading by the clock source set at 70 Hz. This is the rate at which frames of 5x2080 are collected.

4.6 Data Preprocessing

4.6.1 Ultrasound echoes

For each and every 5x2080 frame, we do the following steps of preprocessing:

1. **Bandpass Filtering:** The ultrasound range of interest is centered on 5MHz with a bandwidth from 2MHz - 8MHz. We bandpass the the raw ultrasound echoes with these parameters.

2. **Normalization per channel:** each individual channel (or element) is normalized i.e normalize first row (1x2080), then normalize second row (1x2080) etc. It is also possible to normalize the whole matrix directly. In this case, the relative peaks between elements would be preserved. From our preliminary testing, we found that normalizing per channel yielded better results, since it made the movement of amplitude peaks more defined. Lastly, it also possible to normalize across the entire dataset and use this reference value for new unseen values. These different normalization approaches can be a topic of future study.

3. **Amplitude Envelope:** In traditional ultrasound imaging, key information is contained in the the amplitude envelope of the signal, which relates to the "darkness" of a pixel. The amplitude is computed via the Hilbert transform. Although we are not imaging, we reason the amplitude envelope should contain the same useful information to a machine learning model compared to the raw waveform. In the case of ultrasound echo data, it is the location and morphology of peaks that matter. This is to be contrasted with audio and speech, which often analyzes from the frequency-time domain. The focus on amplitude envelopes is similar to onset detection and voice-activity detection in digital signal processing. Alternatively, one can also do half-wave or full-wave rectification with filtering to achieve the same effect. The latter approach is especially useful in embedded system, where the number of clock cycles per preprocessing step

is of greater concern.

Mean filtering can also be applied to reduce the dimensionality of the data. During the prototyping phase, we mean filtered by a factor of 20, bringing a processed frame of data to 5x200. Linear regression methods and simple fully connected neural network trained on this data could be trained in a couple of minutes on the CPU of a computer. Some of these are implemented as discussed as baseline approaches. However, mean filtering removes granularity in the location of the peaks. For example, a peak that moves within 20 samples will be registered as a single stationary peak with a mean filter window of length 20. Thus, these models were unable to reliably detect intermediate values and could only "jump" between flexion or open hand. In this thesis, we opt to train end-to-end directly on the 5x2080 to preserve the full resolution of the peaks.

4.6.2 Angle data

The raw angles in the X, Y, Z dimensions outputted by our sensor normalized from 0 - 180 to 0.0 - 1.0.

4.6.3 Ground Truth Labels

The raw labels are normalized from 0 - 99 to 0.0 - 1.0.

CHAPTER 5
EXPERIMENTAL APPROACH AND MODEL ARCHITECTURES

5.1 Convolutional Neural Networks (CNN)

In the last five years, Convolutional Neural Networks (CNN) have shown remarkable results in the areas of image classification and image segmentation [21, 22, 23, 24]. A much more comprehensive explanation of CNN's can be found in a recently published book on deep learning by Goodfellow et al. [25]. In the machine learning community, a popular and indicative benchmark of performance is the ImageNet challenge [26], a large database of over 14 Million hand annotated images divided into 1000 distinct classes. Starting in 2010, the annual ImageNet Large Scale Visual Recognition Challenge (ILSRVC) is an annual competition where different research groups benchmark different models on this common dataset. In 2011, a 25% error rate was considered good performance. In 2012, the first deep convolutional neural network achieved an error rate of 16% [21], igniting much of the modern deep learning boom. By 2017, the the majority of competing teams regularly score less than a 5% error-rate, an impressive feat considering human performance is pegged at a 5% error-rate.

An important property owing to a CNNs ability generalize is spatial and translational invariance. The same learned filters are fixed and applied to different parts of the image, forming a hierarchy of abstractions at each layer.

5.2 Multi-modal learning

The advantage of Deep Learning lies in the ability to modularize and combine features from different modalities, such as those from speech, language and vision [27, 28]. These approaches have led to breakthrough in robotics and autonomous vehicles. In the literature,

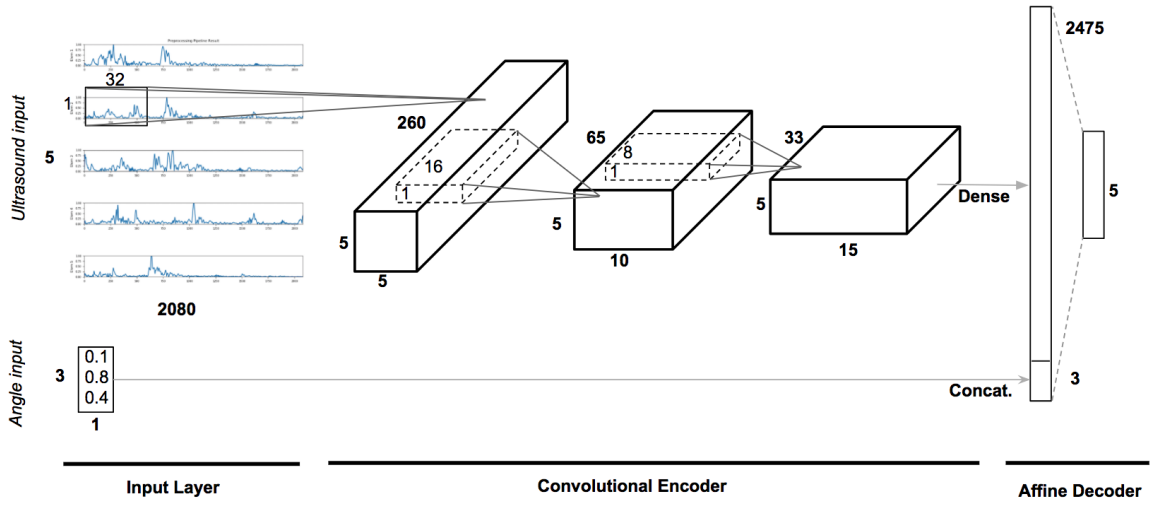


Figure 5.1: CNN model architecture for SEUS transducers

features extracted from different modalities are often "concatenated" into a longer vector, which are then processed and interpreted by subsequent layers in the network [29]. We adopt a similar approach, in which the orientation data is concatenated with later layers of the network.

5.3 Model Architecture

We first describe the final model architecture and motivations behind the design, followed by a discussion of how we arrived at this particular topology and failure cases.

5.3.1 Architecture Design

The model employed in this thesis is a 3 layer CNN, followed by a fully connected layer that is concatenated with the angle input. Each layer contains a dropout of 0.3. A final layer outputs a real-valued 5 dimensional vector. Each layer uses a rectified linear unit (RELU) as an activation function [30]. Only the last layer uses a sigmoid in order to regress values from 0.0 to 1.0. The model architecture is summarized in table 5.1 and Figure 5.1 and is called "SEUS-CNN" for the remainder of the thesis.

Unlike the 2 dimensional convolutions typically employed in deep learning image tasks,

Model Layers	Specification
US Input	(n,5,2080,1)
Conv2D BatchNorm Dropout	Filters=5, Kernel=(1,32), Strides=(1,8), RELU - 0.3
Conv2D BatchNorm Dropout	Filters=10, Kernel=(1,16), Strides=(1,4), RELU - 0.3
Conv2D BatchNorm Dropout	Filters=15, Kernel=(1,8), Strides=(1,2), RELU - 0.3
Concat. Fully Conn.	Conv Features and Angle Input 5, Sigmoid
Total Param.	$\sim$14,000

Table 5.1: Model architecture employed throughout this thesis

we use a 1 dimensional convolution like those employed in digital signal processing. These can be though of "rectangular" filters of size e.g 1x16 or 1x32. The choice of increasing filter number, decreasing filter size and decreasing stride length as the network deepens follows a design pattern employed by Mnih et al. [31] in a deep reinforcement learning system that could play Atari games with super-human performance. We chose the starting filter size of 1x32 because this is approximately the width of a peak in the ultrasound echo. By using a filter of this size with large strides, the model can capture the overall distribution of these peaks across the signal. The smaller strides deeper in the network enable the model to become increasingly sensitive to small shifts in the features extracted by intermediate layers corresponding to these peaks.

However, in the same way CNNs enforce spatial and translational invariance across the image, we would like the rectangular filters in our models to enforce feature invariance across sensors i.e. the same filters learned on sensor element 1 should be useful in interpreting sensor element 4 etc. Unlike the layer of square volumes and feature maps outputted from an image task, the approach taken in this thesis produces rectangular volumes. One of these dimensions remains constant at 5, corresponding to the number of sensors employed. We reason that enforcing these invariances across sensors will encourage the network to generalize to new unseen sittings. Moreover, we hypothesize these learned filters will be useful to a new configuration of sensors, say 10. The learned kernels could be applied to the new set of sensors and the feature maps expanded accordingly. Only the last few layers of the network would need to be re-trained.

The models are trained using Mean Squared Error as a loss function using the Adam Optimizer. We use early stopping to prevent over-fitting.

5.3.2 Division of user-invariant and user-dependent features

We enforce an architecture design that divides the model into a set of user-invariant features in the form of a convolutional encoder (Figure 5.1), and a set of user-dependent features,

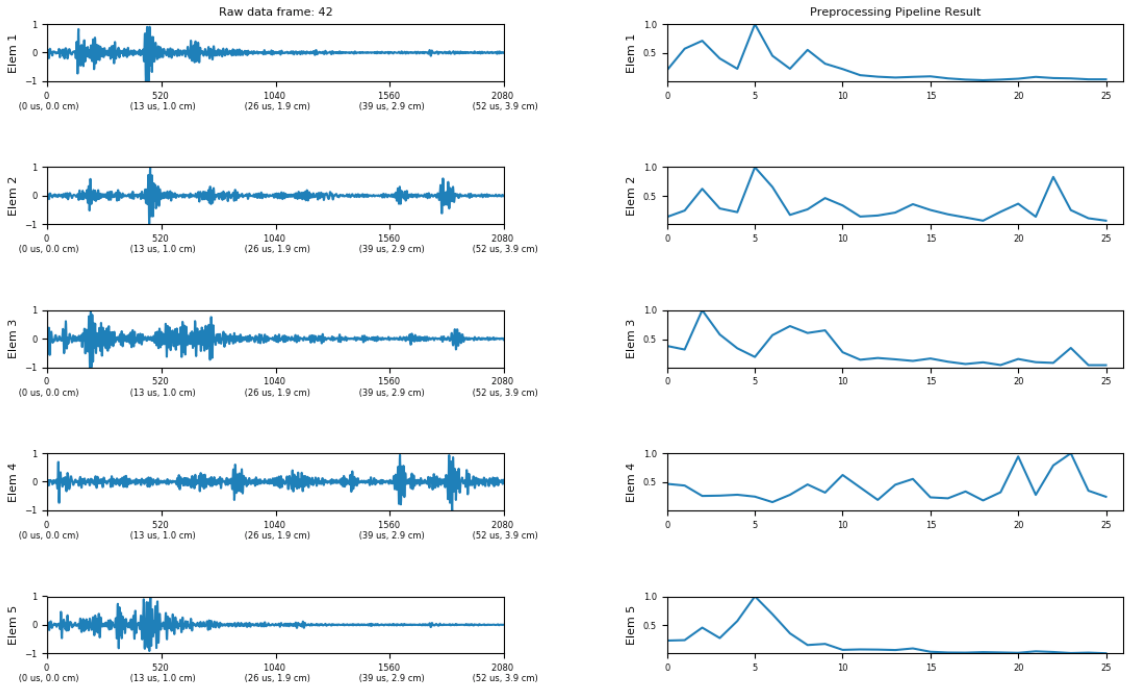


Figure 5.2: Heavy mean filtering on the data to reduce dimensionality during early model design. To be contrasted with figure 4.12

in the form of the affine decoder. We reason that learned filters from a group of users can be transferred to a new unseen user to speed up the process of training. This is the topic of further discussion later.

5.3.3 Process of architecture design

For completeness, we describe failure cases and the process of arriving to the architecture described above. Initially we mean filtered the output by factors in the range of 50 to drastically reduce the data dimensionality. The outcome of this process is shown in Figure 5.2. This enabled models including Linear Regression and a shallow fully connected network to begin predicting output flexions. However, Linear Regression are limited by how the models are unable to simultaneously output a 5-dimensional finger flexion vector unless mathematical modifications beyond the scope of this thesis are employed. Thus a separate model must be trained per finger and assumes finger independence i.e. a linear regression model is trained for the thumb, index, middle etc. and predictions are made per-finger.

For this reason, we moved onto a simple fully connected network that could output 5-dimensional finger values simultaneously. We call this 5 dimensional output vector a "flexion vector" for the remainder of this thesis. In this approach, the model is able to encode information on the predictions of other fingers into the prediction of a particular finger. Our initial approaches consisted of fully connected models with 2 hidden layers of size 8 and 16 respectively, which terminated in the 5 nodes of sigmoid output. During preliminary testing, this approach worked more like a classifier, only able to detect finger states with a flexion of 0.0 or a flexion of 1.0 and nothing in between. This was because the data was mean-filtered so heavily that there were no intermediate ultrasound echoes between fully flexed and open hand.

As a result, we slowly increased the dimensionality of the data and increased the number of layers neurons. Once we began operating at a matrix size of approximately 5x1040, we switched to a convolutional model to enforce generalization, progressing towards the full 5x2080 to retain maximum spatial resolution of the amplitude peaks.

In the design of the CNN model, we found that models with more than 3 layers exhibited significant over-fitting. We also found that adding additional fully connected layers in the affine encoder stage also caused over-fitting, thus we settled on the concatenated layer being connected directly to the output 5 units of the regression layer.

5.3.4 Time independence

One can argue that our approach should take into account a series of frames when outputting a prediction. It is physically impossible for a finger to jump from open to closed within a few milliseconds. Thus, knowledge of previous flexion vector should be useful in the task of predicting the current flexion vector. This statement is true and should be the topic of further study. In the scope of this thesis however, we reason as a first approach that every distinct hand configuration can be uniquely mapped to a corresponding configuration of ultrasound echoes. This condition of a unique 1:1 mapping also means the trained model

can be used at any desired frame-rate. Although the model is trained on data collected at 70 fps to increase the dataset size, the same model can be deployed at a slower 24 fps during inference on a desktop CPU or weaker, embedded hardware.

5.3.5 Relationship with machine learning and DSP approaches

Readers may notice similarities in the approaches described here with DSP and ML literature. 1D convolutional filters are typically used in speech machine learning applications that work directly on the raw audio. These are implemented in architectures such as WaveNet [32] and other audio classification tasks [33]. 2D convolutions are used when the signal is converted into the spectrogram representation [34]. However, in all these applications, the audio is typically summed and averaged to a mono or single-channel signal representation. In this thesis, the 5 separate transducer channels remain as separate channels.

Similarities also exist with work using machine learning to classify a collection of related signals such as activity detection given readings from a 3-axis accelerometer [35]. However, the approaches diverge in the grouping of signals. In the activity detection example, the 3 dimensions of the accelerometer readings are "stacked" i.e. the input signal has a depth of 3 (like 3 color channels in an image) with an example dimension of 1x512x3. In our approach, our signals are grouped contiguously i.e. the input signal has depth 1 (like a typical audio signal) with an example dimension of 5x512x1. A 10-channel system would have dimensions 10x512x1. This distinction is non-trivial, since our preliminary testing shows that grouping the signals in the second case yielded superior performance as it enforces the desired structure where learned kernels are invariant over channels. The first case enforces the model to learn filters that encapsulate relationships specific to the ordering and number of sensors.

CHAPTER 6
EXPERIMENTAL RESULTS AND DISCUSSION

6.1 Experiment descriptions

We perform two large scale experiments in this thesis:

1. **Regression:** For each user, train a unique model on a single user for regressing finger flexions

2. **Classification:** Threshold the continuous values from the previously trained regression model into discrete class outputs.

6.2 Experiment 1: Regression of Flexion vectors

6.2.1 Task Description

Given an input ultrasound echo matrix of dimensionality 5x2080, learn a function $f(x, \theta)$ parameterized by θ that outputs a 5 dimensional output flexion vector with range 0.0 - 1.0, corresponding to the individual flexions of each finger. In this experiment, the task is defined per user, meaning a single model is trained on data from one user and then tested on an unseen test set from the same user. We motivate this approach as the first experiment due to the clean separation in a of user's echo patterns in a TSNE plot (t-Distributed Stochastic Neighbor Embedding), shown in Figure 6.1 on page 43. Intra-user echoes are highly related, but inter-user patterns exhibit enough differences that a dimensionality reduction technique like TSNE separates them into individual clusters. Moreover, notice how there is no clean separation of fingers within each user. We hope that in the process of learning how to output flexion vectors, the model is able to disentangle these overlapping input echoes.

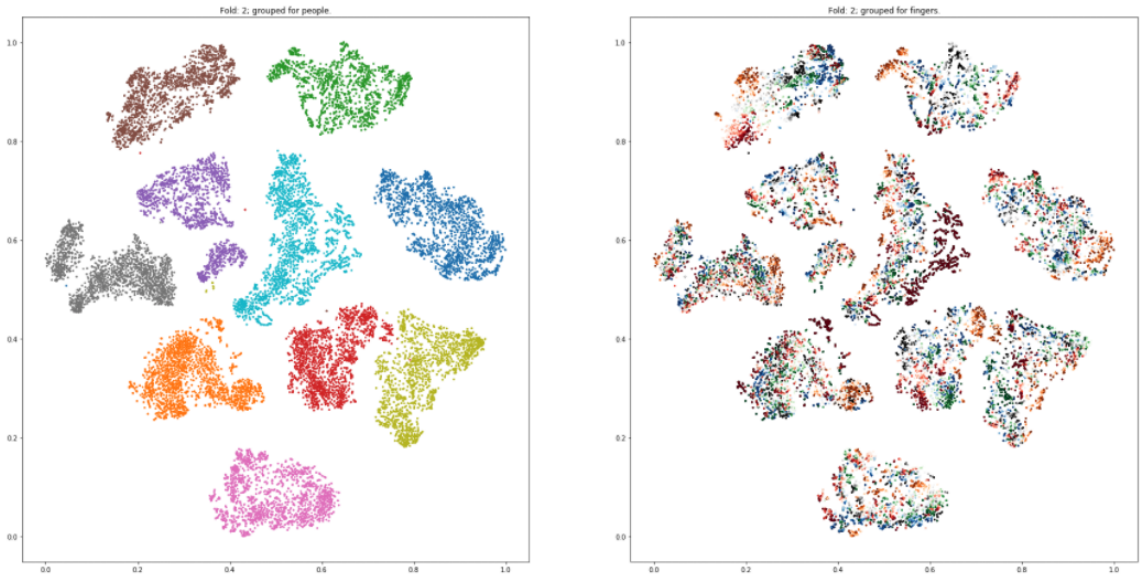


Figure 6.1: TSNE plot of datapoints from 10 users. **Left:** Datapoints colored by user (i.e. 10 separate clusters = 10 separate users) **Right:** The same datapoints, but colored by finger, with color intensity signifying flexion strength. The group clusters are identical to the left TSNE plot. Note how within a user, each finger is mixed with other fingers.

6.2.2 Division of Train, Validation and Test Sets

We perform 5-fold cross validation across the 5 sessions of data collected per user. For example in fold 1, we completely remove session 1 (associated with sitting 1 and sitting 2) from the dataset. We then randomly permute data from session 2,3,4,5 and perform a training and validation split of 95% and 5%, yielding a training set of approximately 350,000 data points and a validation set of 50,000 data points. This process is depicted at the top of Figure 6.2 on page 44. The 100,000 training data points from session 1 is an unseen test set. This configuration is the most realistic, since the trained model has never seen data from testset sitting and simulates a user who has just put the sensor on. We report results for this configuration as a test for generalization across sensor shifts.

We note it is incorrect experimental protocol to first group all sessions 1,2,3,4,5 together, randomly permute the aggregated dataset and then divide into training, validation and testing set. This process is depicted at the bottom of Figure 6.2 on page 44. This approach violates the independent and identically distributed (IID) condition. The ultrasound echoes are a

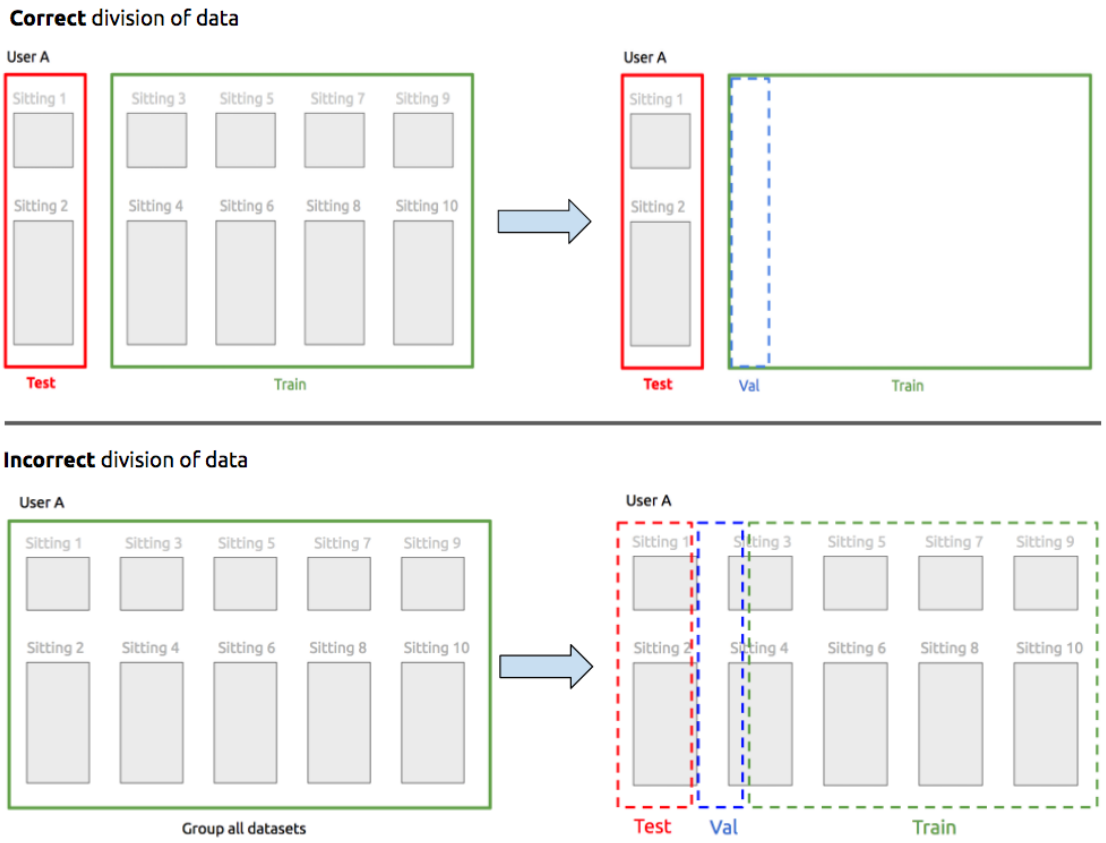


Figure 6.2: **Top:** Correct division of dataset. During a single fold for a user, two sittings are reserved as a test set and completely removed. The training data is then partitioned into a training and validation set. In this scenario the model has never seen data from the sensor location of the test set. **Bottom:** Incorrect division of dataset. All sittings are grouped into one large dataset, and then divided into a testing, validation and training set. In this scenario, the test set contains many different, but highly correlated samples with the training set.

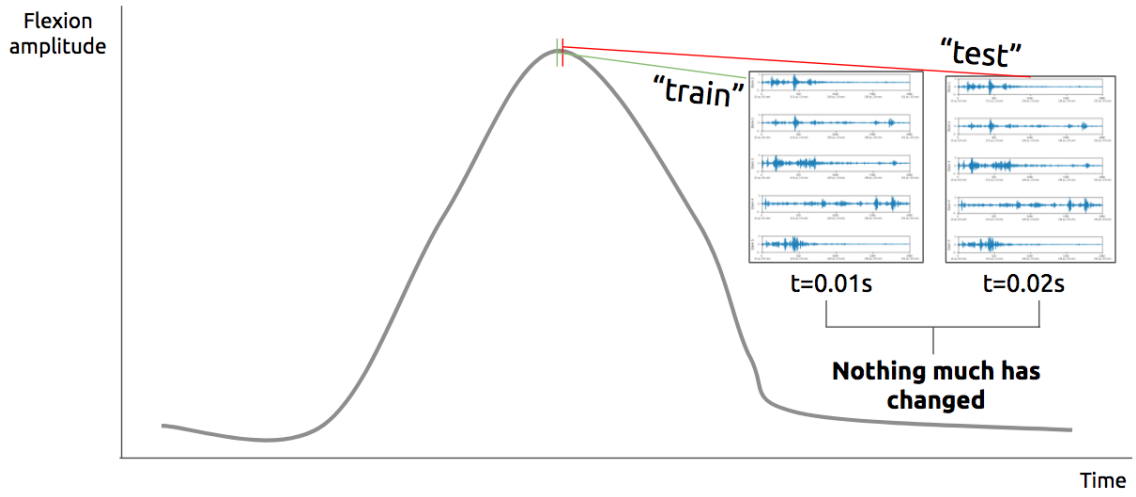


Figure 6.3: Diagram depicting a flexion of one finger. At the apex, a sample is randomly assigned to the training set and a neighboring sample is assigned to the test set. However, due to the high sampling rate of 70Hz, nothing much has changed between these two contiguous samples. Thus, results may be inflated due to the high correlation between samples in the training and test set.

time series data, sampled at a high rate of 70Hz. At the same time, an individual finger flexion occurs over the course of 2 - 6 seconds. Thus, there are many consecutive frames where nothing has changed between samples, as shown in Figure 6.3 on page 45. When these individual data points are randomly permuted, it is likely for a sample to be placed in training, and the neighboring sample to be placed in the testing set. These two samples are highly correlated and in certain cases, resembles a regime where the model has memorized a training sample and encounters the same sample in the testing set. Results reported from this method would thus be inflated. For this reason, we enforce the data division outlined above, in which a session is completely removed from the fold training.

It is unclear whether previous work in the field follow this protocol. We thus also report results on the validation set, which reflect the correlated of data described here, in conjunction with results from the test set.

We still favor a high sample rate for several reasons. From our experiments, we notice how ultrasound echoes contain low amplitude noise in the input signal (below 0.1 normalized amplitude). Occasionally, interference from other devices and the environment also

introduce signals at approximately 0.3 normalized amplitude. By sampling data at a high rate, we capture these small variabilities and can train the model to be robust against this signal noise. Although it is possible to synthetically augment the data in this manner from data collected at a lower frame rate, it is more realistic to collect data from the natural distribution of noise found in the physical system.

6.2.3 Dataset Imbalances

Due to the nature in which the dataset is collected, where the user pauses in open hand position between each flexion to move to the new designated orientation, there is an overwhelming number of open hand samples. The dataset is balanced by randomly under-sampling the open hand data points during each fold. We note that in the case of ultrasound images, this does not seem to affect the ability for the model to do regression. However, in the case of ultrasound echoes, a dataset imbalance favoring open hand causes the model to ignore ring and pinky fingers. This is confirmed by visual inspection: the ultrasound echo patterns for ring and pinky fingers look most similar to open hand position, with a few shifts in the characteristic pulses. This is to be contrasted with thumb and index fingers, where a flexion causes a large and noticeable change in ultrasound echoes.

6.2.4 Baseline Implementation

As a baseline, we implement Linear Regression on preprocessed ultrasound echoes. We perform an additional mean filter and reduce the input matrix to a size of 5x104. This number of dimensions is a tradeoff between two extremes of dimensionality: the full 5x2080 matrix is too high for regression while a matrix of 5x20 removes too much granularity from the data. Since Linear Regression can only output 1 dimension of continuous data at a time, we train 5 separate linear regression models, one for each entry in the 5 dimensional flexion vector corresponding to each finger. The model for a particular finger is trained on balanced data for that finger. From the author's understanding, this is the same approach described in

previous regression work by Castellini et al. [14].

6.2.5 Metrics

We report 5 metrics for regression performance and briefly explain how they should be interpreted in relation to the experimentation:

- $\boldsymbol{R^2}$: This value should be regarded as the strongest metric for evaluating performance. Perfect performance is denoted by a value 1.0. It is possible for $\boldsymbol{R^2}$ to go arbitrarily negative, signifying worse and worse regression performance.

- **Pearson Coefficient (P.Coeff)**: Denotes the correlation between two sets of data, with 1.0 meaning strong positive correlation and -1.0 meaning strong negative correlation. In our case, the two sets of data are the predicted and true values correspondingly. Although $\boldsymbol{R^2}$ should be regarded as the main metric, we include the Pearson Coefficient for completeness.

- **Mean Absolute Error (MAE)**: Denotes the absolute error between the labeled value and the model's predicted value. The range of labeled values is normalized 0.0 - 1.0, thus the MAE is a direct, absolute measurement of how large an error the model will make during regression. However, we note that MAE should always be used in conjunction with $\boldsymbol{R^2}$, since it may not reveal situations where the model is predicting a constant output flexion vector for all inputs.

- **Mean Squared Error (MSE)**: Denotes the squared error between the labeled value and the model's predicted value. Although readers should regard MAE as a more direct measurement of model error performance, we include this metric since it is used as the loss function to train the network. The MSE results reported here reflect the average loss achieved by the model at the end of training.

- **Root Mean Squared Error (RMSE)**: Denotes the root of the mean squared error

between the labeled value and the model's predicted value. We include this for completeness.

6.2.6 Results

Only the main results of interest are shown here. All metrics are averaged over 50 folds across 10 users. Comprehensive results for each individual user are included in the Appendix. We report that inference for all results in this section can be done in real-time up to the original data sampling rate of 70Hz on the CPU of a computer.

Table 6.1 on page 49 shows metrics for the SEUS-CNN's regression performance. We report the metrics per finger and a combined value reflecting performance across all fingers. The combined value is an unweighted average of the metrics from the individual fingers. Table 6.2 on page 49 shows metrics for the SEUS-CNN on the validation set. Note that for metrics like R^2 and Pearson Coefficient, higher is better. However, for metrics such as MAE, MSE and RMSE, lower is better.

Similarly to the SEUS-CNN, Table 6.3 on page 49 shows metrics for Linear Regression. Table 6.2 on page 49 shows metrics Linear Regression on the validation set.

Overall, for the SEUS-CNN we report an average R^2 of 0.632 and a average MAE of 0.094 on the test set. For Linear Regression, we report an average R^2 of -2.562 and an average MAE of 0.388 on the test set.

6.2.7 Discussion

Test set

Referring to table 6.1 on page 49, we report a combined R^2 of 0.632. Given these metrics are averaged over 10 users and across different arm orientations, these are promising results underscoring the ability to perform regression of finger flexions using single-element transducers and CNN's. The baseline linear regression model on the other hand, perform very poorly with negative R^2 values in all fingers and combined R^2 of -1.747. The SEUS-CNN's

Metric/Finger	Thumb	Index	Middle	Ring	Little	**All**	Remarks
R^2	0.723	0.684	0.619	0.523	0.613	**0.632**	Higher is better
P.Coeff	0.85	0.827	0.795	0.723	0.783	**0.796**	
MAE	0.069	0.085	0.095	0.125	0.094	**0.094**	Lower is better
MSE	0.025	0.029	0.037	0.048	0.034	**0.035**	
RMSE	0.158	0.170	0.192	0.219	0.184	**0.187**	

Table 6.1: SEUS-CNN Test Set Results. Metrics averaged over 50 total folds across all 10 users with ∼3,000,000 samples

Metric/Finger	Thumb	Index	Middle	Ring	Little	**All**	Remarks
R^2	0.711	0.658	0.611	0.493	0.592	**0.613**	Higher is better
P.Coeff	0.843	0.811	0.79	0.702	0.769	**0.783**	
MAE	0.071	0.089	0.097	0.127	0.096	**0.096**	Lower is better
MSE	0.026	0.032	0.038	0.05	0.036	**0.036**	
RMSE	0.161	0.179	0.195	0.223	0.189	**0.189**	

Table 6.2: SEUS-CNN Validation Set Results. Metrics averaged over 50 total folds across all 10 users with ∼1,000,000 samples

Metric/Finger	Thumb	Index	Middle	Ring	Little	**All**	Remarks
R^2	-1.582	-1.968	-2.091	-3.689	-3.479	**-2.562**	Higher is better
P.Coeff	0.308	0.311	0.309	0.147	0.165	**0.248**	
MAE	0.328	0.352	0.359	0.458	0.441	**0.388**	Lower is better
MSE	0.171	0.194	0.201	0.309	0.288	**0.232**	
RMSE	0.414	0.440	0.448	0.559	0.537	**0.481**	

Table 6.3: Linear Regression Test Set Results. Metrics averaged over 50 total folds across all 10 users with ∼3,000,000 samples

Metric/Finger	Thumb	Index	Middle	Ring	Little	**All**	Remarks
R^2	-1.089	-1.398	-1.160	-2.784	-2.304	**-1.747**	Higher is better
P.Coeff	0.444	0.420	0.458	0.264	0.305	**0.378**	
MAE	0.288	0.308	0.296	0.403	0.371	**0.333**	Lower is better
MSE	0.137	0.156	0.140	0.250	0.212	**0.179**	
RMSE	0.401	0.395	0.374	0.500	0.460	**0.423**	

Table 6.4: Linear Regression Validation Set Results. Metrics averaged over 50 total folds across all 10 users with ∼1,000,000 samples

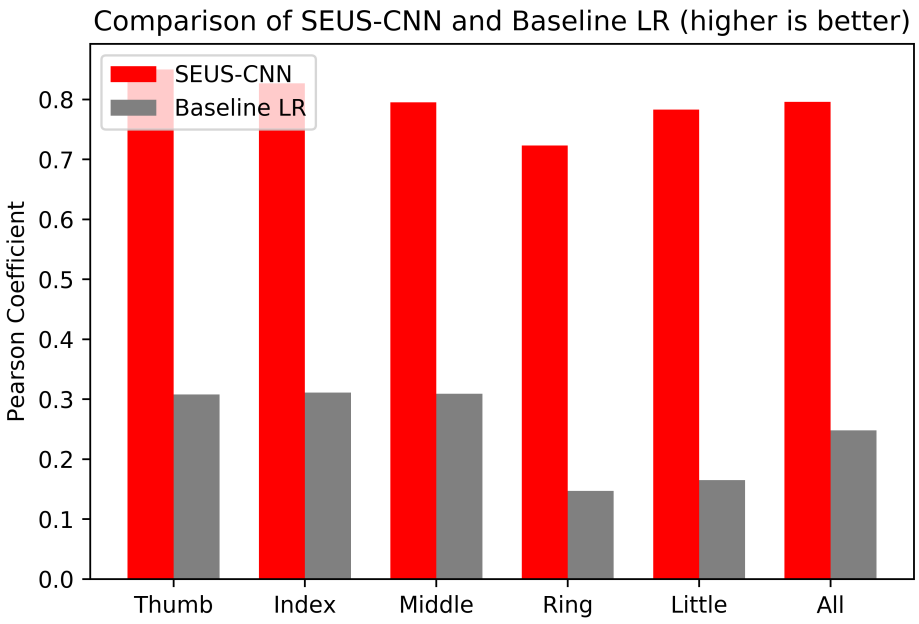


Figure 6.4: Comparison of SEUS-CNN with baseline linear regression in terms of Pearson Coefficient per finger and overall score. The SEUS-CNN outperforms the baseline in all fingers. Note that higher is better.

better performance is reflected in a higher Pearson Coefficient as shown in Figure 6.4 on page 50 and lower overall errors as shown in Figure 6.5 on page 51

The reason for the poor performance in linear regression is because there are 5 independent models outputting 5 independent regression predictions. Upon visual inspection of the model output over time, we can see why this is the case. Observe how in Figure 6.6 on page 52, linear regression is actually able to quite reliably regress the thumb values. In this figure, the five consecutive subplots show the thumb, index, middle, ring and pinky fingers respectively. However, the linear regression models associated with the other fingers are unable to output the correct flexion value of 0.0 when the thumb is flexed. Instead, models for the other fingers are also activated incorrectly, almost in phase with the thumb. When the error is calculated across the entire flexion vector, the incorrect value across all the other 4 fingers are summed and incur a very high error.

One could argue that this is unfair towards the linear regression model. Instead, one could measure the thumb performance on thumb samples, index performance on index

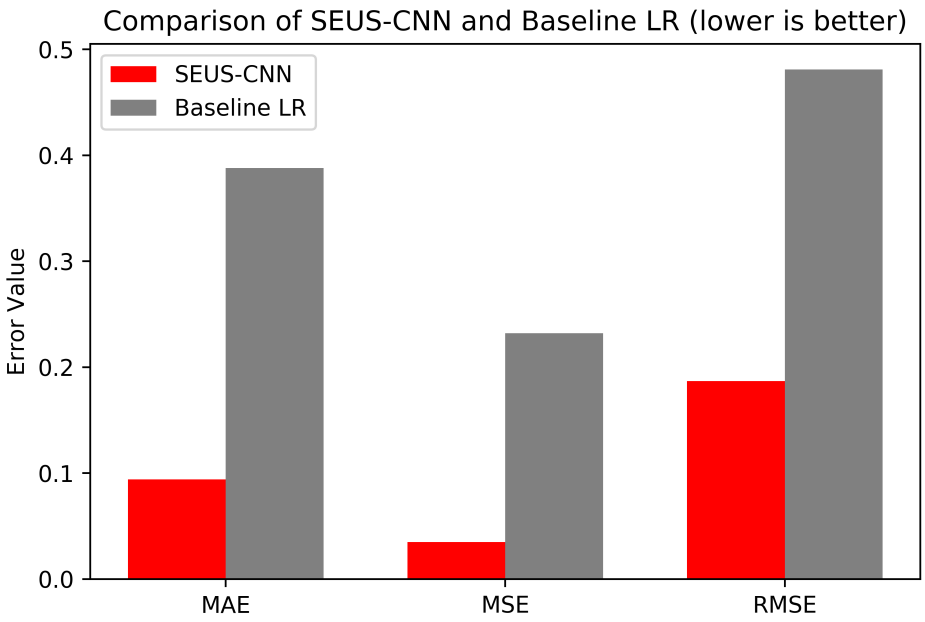


Figure 6.5: Comparison of SEUS-CNN with baseline linear regression the average error achieved in terms of overall MAE, MSE and RMSE. Note that lower is better.

samples etc. However this is unrealistic, since it suggests a hand is only ever composed of one finger and never all five fingers at once.

Contrast this to the output of our SEUS-CNN model shown in figure 6.7 on page 52. For a similar input of only thumb flexions, the SEUS-CNN is able to regress the correct thumb value and values equal or close to 0.0 on all the other open fingers as a concurrent 5-dimensional vector per timestep. The little finger for example, remains consistently at 0.0 throughout the period of 21 seconds, whereas in Linear Regression, the pinky values, as well as other finger regressions, seem to cycle in phase with the thumb values. This indicates a linear combination of the ultrasound echo feature space is unable to disentangle the overlapping input samples shown earlier in the TSNE plot (figure 6.1).

Alternatively, one could feed the thumb linear regression model all the other fingers as negative examples. Whilst this may address the shortcomings of 5 independent linear regression models, it would require a non-trivial sub-sampling of negative examples. Training a thumb linear regression model to output 0.0 on index, middle, ring and little finger samples

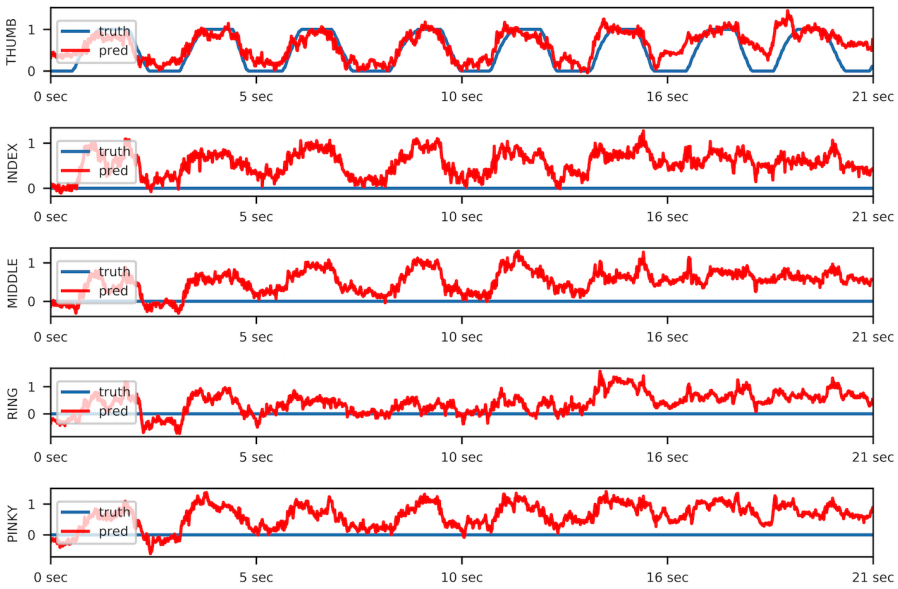


Figure 6.6: Output of Baseline Linear Regression on thumb movement. In this figure, we concatenate 21 seconds (21 * 70 = 1470 frames) of flexion vectors per timestep. Note how linear regression is able quite reliably regress the thumb flexions. However, the linear regression models for other fingers fail to output a 0.0 value.

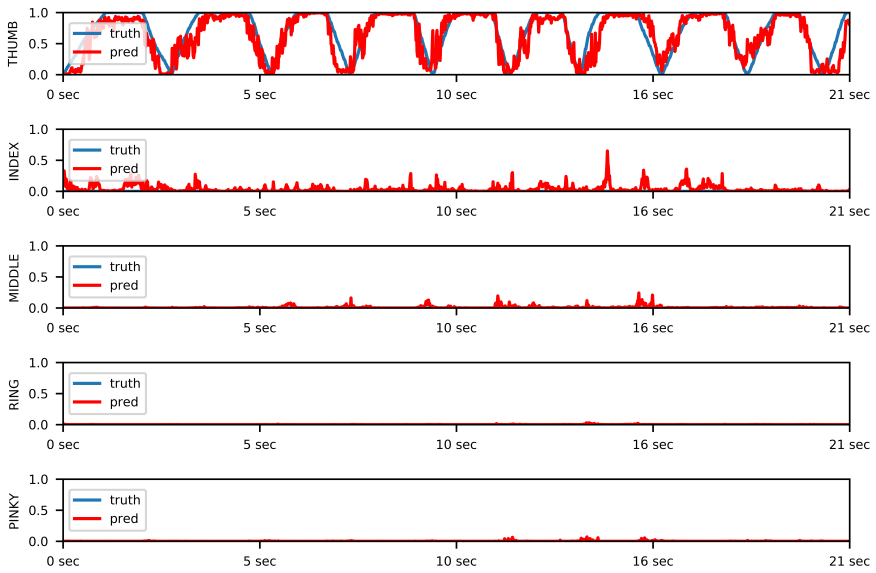


Figure 6.7: Output of SEUS-CNN on thumb movement. Note how for a similar input to the Linear Regression model in figure 6.6, the SEUS-CNN is able to correctly regress the thumb movement and output 0.0 for all other fingers as a simultaneous 5-dimensional vector per timestep

would mean the dataset now contains 4 times more samples labeled as 0.0 than there are positive samples of a thumb flexion from 0.0 - 1.0. Additional precautions would need to be taken to ensure how sub-sampling the negative examples yields an equal distribution across the other fingers. We leave this, as well as other possible multivariate regression methods as an experiment for a future study. For now, we highlight how the SEUS-CNN is able to more robustly regress the flexion vectors compared to the baseline

A closer qualitative examination of model output per timestep

Figure 6.8 on page 54 shows an example of a trained model with a particularly high R^2 value (approximately 0.90) regressing a thumb flexion on the test set. In these figures, we concatenate a shorter period of 4 seconds of subsequent frames (or $4 * 60 = 240$ frames) to show how the model regresses intermediate values. This is easier to discern in Figure 6.8 than the longer time period of 21 seconds used previously in Figure 6.6 on page 52 when comparing agains the baseline. The last plot shows the angles X, Y, Z recorded by the accelerometer as the user moves their arms to different orientations. We remind the reader that the model is only regressing the finger flexions, not the angle outputs.

The model is able to regress the thumb flexion vector while the orientation is changing, as recorded by accelerometer is changing between these flexions (shown at the very bottom of the figure). At 3 seconds, the model outputs a slight activation on the index. It is unclear whether the model is outputting prediction crosstalk, or is truly modeling the underlying physiology of wrist tendons. These can only be validated through examination of the corresponding synchronized ultrasound image data discussed later in the chapter.

In Figure 6.9 on page 55, the model is regressing on a middle flexion. Note how through visual inspection, we can see the model is qualitatively performing worse than the thumb. It is less able to regress intermediate values, for example dropping off back to 0.0 at a time of 1 second when the flexion actually lasts longer. This qualitatively worse performance is captured by metrics such as R^2.

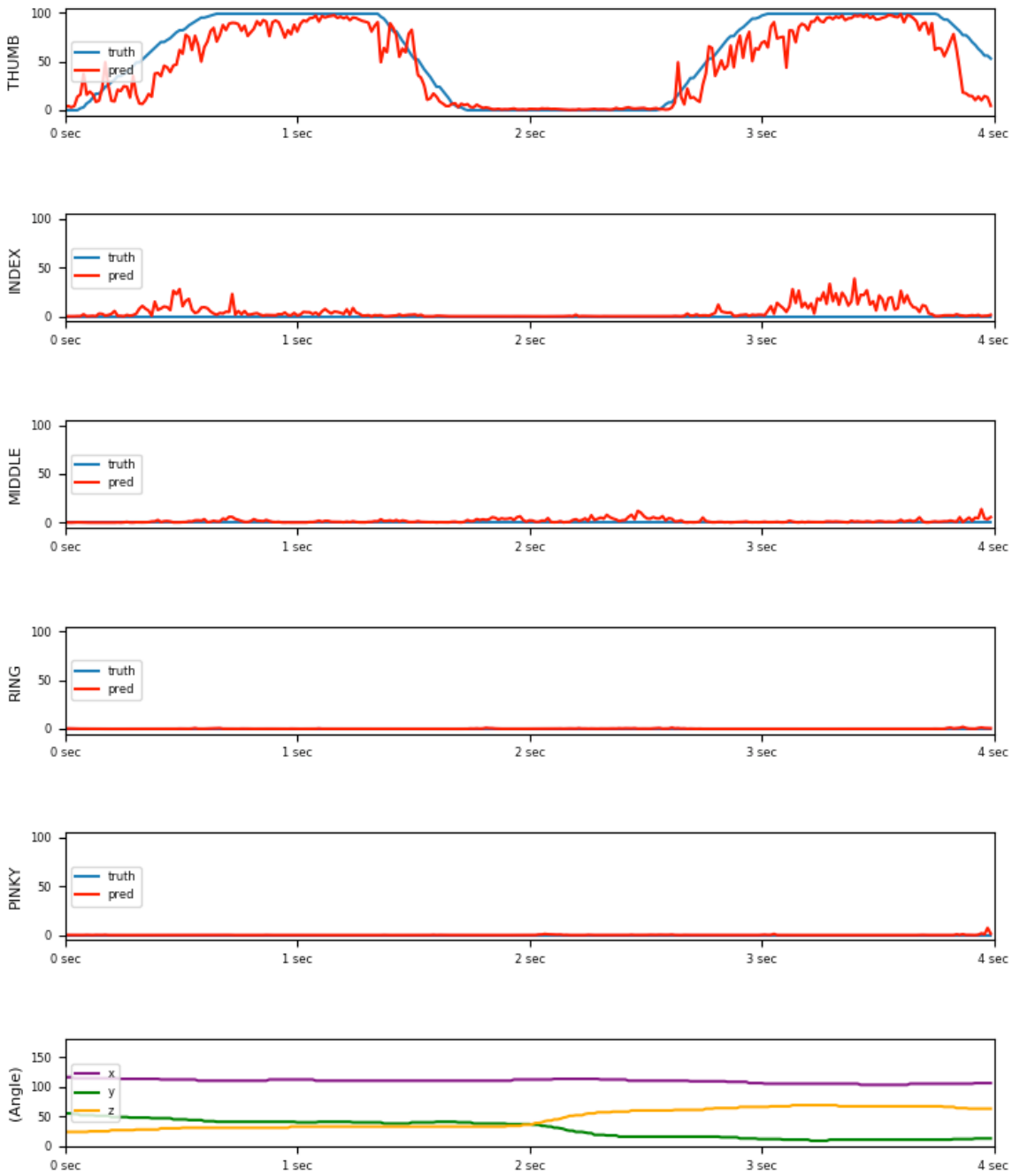


Figure 6.8: Example regression of trained model on a thumb flexion from the Holdout Test set. In this figure, we concatenate 4 seconds of subsequent frames (or $4 * 60 = 240$ frames) to show model prediction across time. The bottom of the figure shows the X, Y, Z accelerometer readings as they change over time. Note how the first thumb flexion from 0-2 seconds is done in one arm orientation, followed by a second flexion in a different orientation

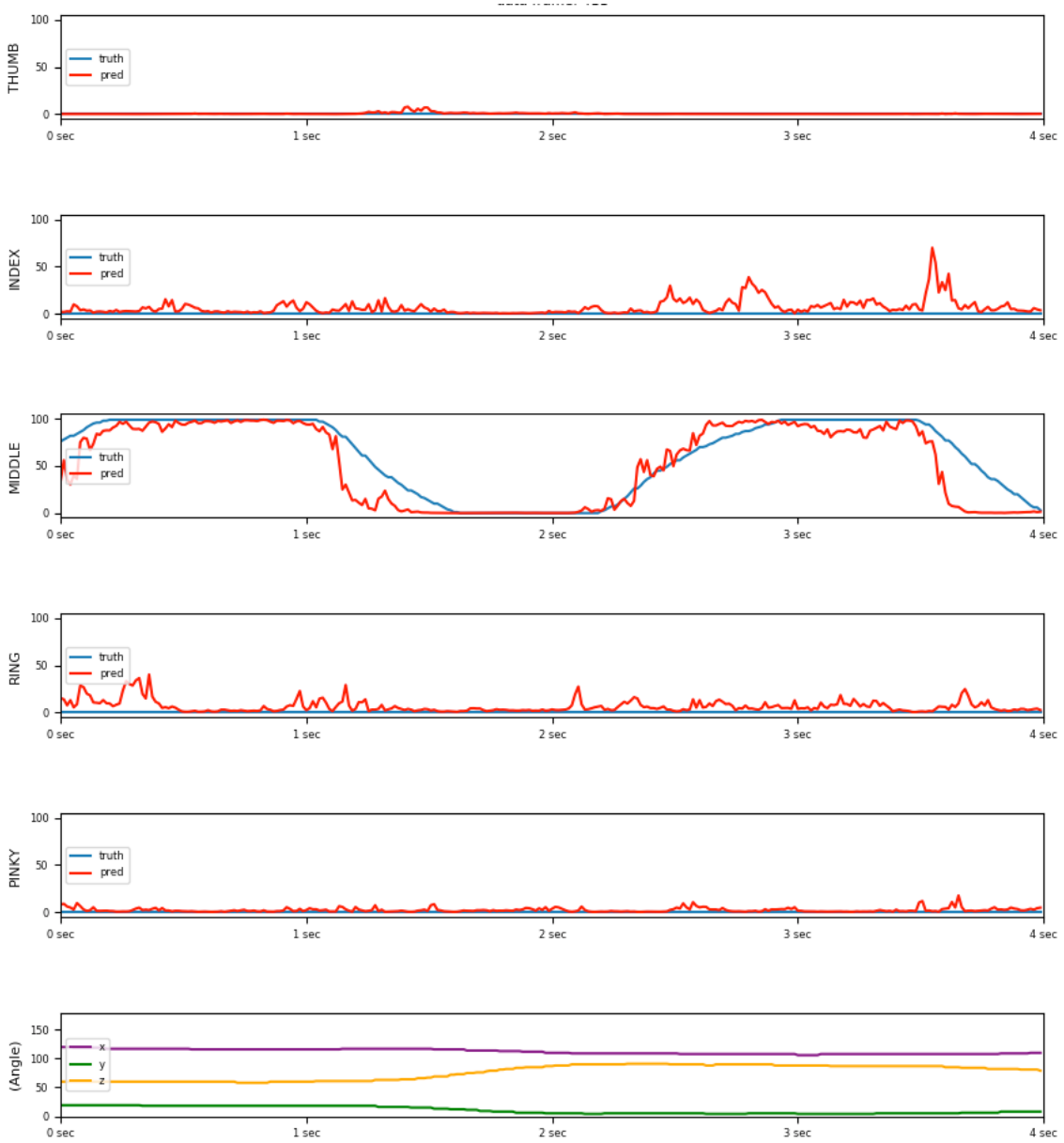


Figure 6.9: Example regression of trained model on a middle flexion from the Holdout Test set. Note how middle finger flexions cause associated activations in the ring finger and index finger

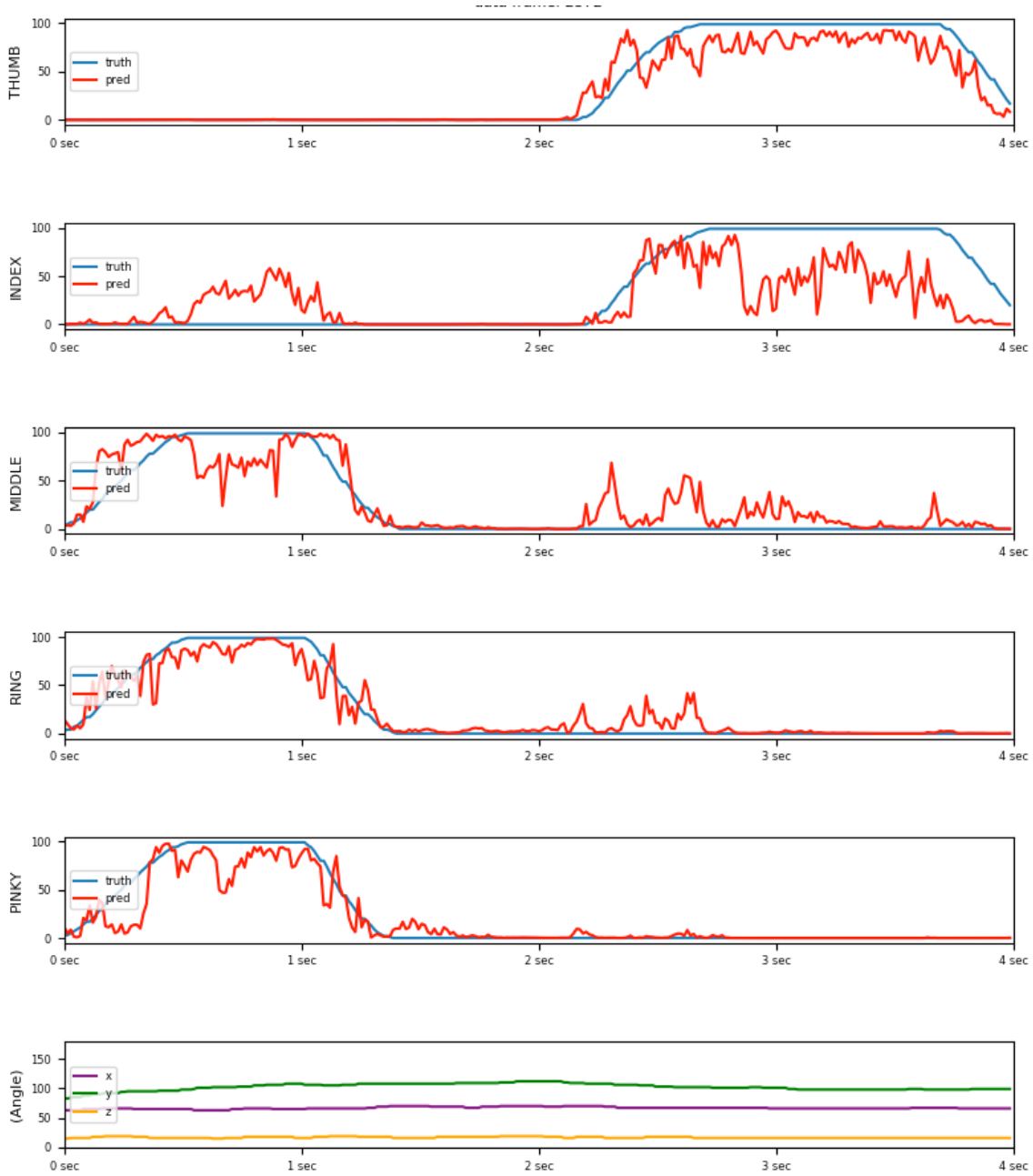


Figure 6.10: Example regression of trained model on a mixed flexions from the Holdout Test set. These show the user curling their middle, ring and pinky fingers simultaneously (like the hand gesture for "gun") and then curling their index and thumb right after (like the hand gesture for "ok"), see figure 4.1 for pictures of these actions. Considering the same model is regressing both individual and simultaneous finger regressions, these are promising first results.

In Figure 6.10, we see the performance of the model on mixed gestures. Regressing a free and simultaneous finger flexions is a very challenging task not previously addressed by previous work. This figure demonstrates a user curling their middle, ring and pinky fingers (denoting a "gun" hand gesture) and then curling their thumb and index fingers (denoting an "ok" hand gesture). Some of these actions were depicted earlier in Figure 4.1 on page 16. We emphasize how figures 6.8 through 6.10 of various flexion vectors including single and simultaneous flexions are all made by the same model. These plots provide exciting evidence that it is possible for a single model to output finger regressions whilst being robust to different arm orientations.

The model is able to regress both individual and simultaneous flexions. We stress how electromyography (sEMG) – the current standard for interfacing muscles with devices such as prosthetics or technology – cannot regress finger by finger and mixed flexions simultaneously as demonstrated here. Robust sEMG implementations are limited to discrete classification of a collection of gestures [36].

However, we acknowledge that regression of simultaneous flexions is generally worse than the single finger flexion case. There are many possible explanations, including how 5 sensors may be insufficient for this task or the model architecture's capacity may need to be increased with deeper layers. We discuss these in the next section.

Hypotheses behind model performance

Looking at R^2 for individual fingers shows the best performing finger is the thumb (R^2=0.723), while the worst performing finger is the ring finger (R^2=0.523). The "performance" of a model is highly dependent on three factors: the machine learning model, the hardware configuration and human physiology. Any one of these factors, or all of them, could be affecting the metric. We provide some possible explanations under these different headings, motivating them as hypotheses that should be tested in future work:

- **Physiological:** We hypothesize that the ring finger's relatively worse performance

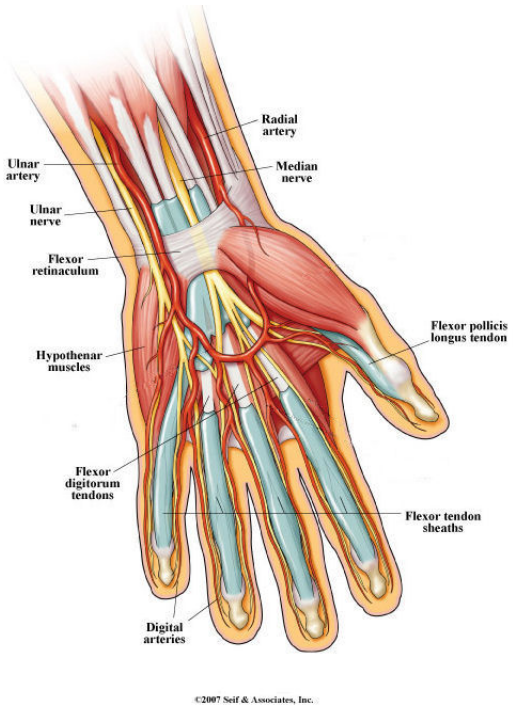


Figure 6.11: Cross section of wrist, showing tendons and nerves. Note the tied tendons from the middle and ring fingers. Image from `https://secure.familyhealthtracker.com/`

maybe related to how it has tied tendons with the middle finger. Figure 6.11 on page 58 shows a cross section of the wrist. Note how the tendons (shown in white) between the middle and ring fingers are in close proximity near the location of our sensor placement compared to the tendons for the thumb, index and pinky, which are more clearly separated. A motion in the middle finger will invariably show movement in the same wrist tendons and muscles, which would cause our system to confuse the two. We notice that a better performing ring finger is often accompanied by a drop in performance on the middle finger.

To validate this hypothesis, a future study involving a physiology expert should leverage the synchronized ultrasound images collected in this thesis. The study should examine whether this particular location of sensor placement indeed makes it physiologically difficult to disentangle middle and ring fingers. Moreover, in a broader sense, a similar study of equal scale to the one presented in this thesis should examine

different sensor locations across the arm. There are many possibilities, including dispersing multiple transducers to different locations (e.g a sensor on the wrist, a sensor on the forearm etc.) or moving the band further up or down the wrist.

- **Ground Truth Data Limitations**: We report an average of MAE of 0.094. It is important to note how there are imperfections in our ground truth data, since it does not account for involuntary finger flexions triggered by voluntary ones. It is impossible to flex the middle finger without moving the ring and slightly moving the index fingers (the reader is encouraged to flex their own middle finger and observe the involuntary movements associated with the ring and index fingers). This means the model can't achieve a true MAE of 0.0 in this dataset, which could be affecting the model's ability to learin.

  Nonetheless, there is evidence to support how these involuntary flexions may be detected by the model, indicating that to a certain extent, the model may have been able to correlate features from the ring and index fingers with the middle finger without this information being explicitly encoded in the ground truth. For example in figure 6.9 on page 55, which shows regression of the middle finger, we see activations on the ring and index finger as predicted by the model, but none in the thumb and pinky as one would expect from a physiological perspective.

  We propose a future study that examines and compare how well the model presented in this thesis performs with different ground truths as discussed in section 4.3.3. By comparing this trained model's output ground truth data with say, a data glove, we can determine whether the model's activations in other fingers is indicative of true physiological movement, or is actually unwanted prediction crosstalk. We note this can and should be done using the models already trained in this thesis.

- **Hardware**: Our approach is centered around 5 single element transducers arranged in a horizontal band on a semi-rigid brace. Alternative configurations with more

or fewer sensors may yield superior performance. Moreover, these sensors can be arranged in different topologies, such as laterally up the wrist or even in a "cross" shape, as suggested by the principal investigator in physiology. To facilitate testing of these alternative configurations in a future work, the author recommends using the trained models in this thesis as the pre-trained convolutional layers in these new configurations. As motivated earlier in section 5.3.1 on the model architecture, the network structure has been designed to enforce filter invariance over sensors. Thus the same filters learned over these 5 sensors should be extendible to new configurations.

A future iteration of the hardware should remove the brace completely for a variety of reasons. Firstly, it is closer to the configuration experienced by the end user, in which the user can freely strap on the sensor like a watch. Second, it would enable the user to rotate their wrist during data collection, a challenge that no research has tackled at the time of writing. Third, it would enable more natural finger flexions. These factors enable better quality data to be collected.

- **Model Architecture**: The 3-layer, multi-modal CNN employed in this thesis is an effective first approach to the problem. We arrived at this model during preliminary testing because deeper models tended to over-fit the dataset and performed poorly on a new sensor location. Nonetheless, many exciting ameliorations are possible in the design of the neural network. Alternative convolution types, such as atrous or dilated convolutions may be better suited to this task. These have been employed to 1D audio signals in the speech domain [32] with great success. The authors behind WaveNet, a paradigm-shifting approach in deep learning for audio, argue that dilated convolutions enable to the model to maintain a wider receptive view of the original signal as the model becomes progressively deeper. This ability to retain a wider receptive view is highly relevant to our task at hand, as the network is essentially learning to become a non-linear and adaptive peak detector that can account for peak and pulse morphology across the entire signal. A range of additional changes, such as the use of residual

networks and convolutional-recurrent networks are motivated later in Future Work.

Validation Set

There is strong agreement between the Test set, shown in Table 6.1 on 49, and the Validation set for all fingers, shown in Table 6.2 on 49. The agreement between these two sets indicates homogeneous data. Our high sampling rate means there may be a lot of sample points that look alike in the dataset. The agreement may also indicate the variety of sensor locations and dataset size in this study is adequate to enforce generalization to new unseen locations. When trained with at least 8 slightly different sensor locations and a training set of $\sim$300,000 samples (roughly 1.5 hours of data from a user), the model becomes robust and invariant to small shifts in sensor location associated with a user taking off and putting back on the sensor band. Future work on this topic should explore the minimum number of shifted sensor locations and minimum number of samples to achieve this level of generalization by progressively reducing the of percentage data employed during training.

Comparison with previous work

We attempt to make very rough comparison with the only previous work on finger flexion regression using ultrasound by by Castellini et al. [14] and Gonzalez et al. [1]. However this comparison is difficult since there are many important differences between regression in previous work and our regression implementation. These are listed and explained next:

- **Images vs. Echoes**: Castellini et al. uses a full imaging sensor while we use a minimal 5-element SEUS transducer. There is significantly more information contained in a US image formed from an imaging array with 128 or more transducers compared to our 5-transducer non-imaged system.

- **Stationary vs. Non-stationary operation**: Castellini et al. requires stationary operation during both data collection and inference. Castellini notes how their approach does not account for arm nor wrist rotations if the arm is moved to a new location or

orientation. Contrast this to our work, where our data collection and model inference handles different arm oriented in a variety of locations (but not yet wrist rotations)

- **Individual Flexions vs. Simultaneous Flexions**: Castellini et al. asked participants to only flex individual fingers: thumb, index, middle, ring and pinky. The metrics are evaluated over these separate finger flexions. Contrast this to our approach, where we include a range of arbitrary mixed gestures involving simultaneous finger flexions. Our models are expected not only to regress individual finger flexions, but also simultaneous finger flexions. No previous work has attempted this task.

- **Hand crafted features vs. End-to-End learning**: Castellini et al. uses a set of hand-crafted features known as Regions of Interest or "ROI's" that summarize changes in key areas of the ultrasound image [14]. These were motivated from specialized physiological knowledge and a comparison of different image features. Contrast this to our approach, where the features are learned end-to-end from our Convolutional Neural Network. The only preprocessing steps are minimal transformations standard in ultrasound signal processing, namely bandpass filtering, normalization and amplitude envelope. We do not explicitly extract any specialized features beyond these preprocessing steps.

Castelinni et al. and Gonzalez et al. report RMSE values of approximately 0.02 and 0.03 respectively, depending on the percentage of data employed during training. These are better results than our overall RMSE of 0.187 and point towards a performance drop-off when moving from systems based on US images to SEUS transducers. A future work should apply Castelinni et al.'s ROI method on the synchronized ultrasound image data in order to produce metrics that enable direct comparison with the ultrasound echoes reported here. By using the image-echo dataset collected in this thesis for this experiment, stronger conclusions can be drawn on the regression performance based on US images and ultrasound echoes. It is also unclear how previous work compute this regression metric, since the flexions of other

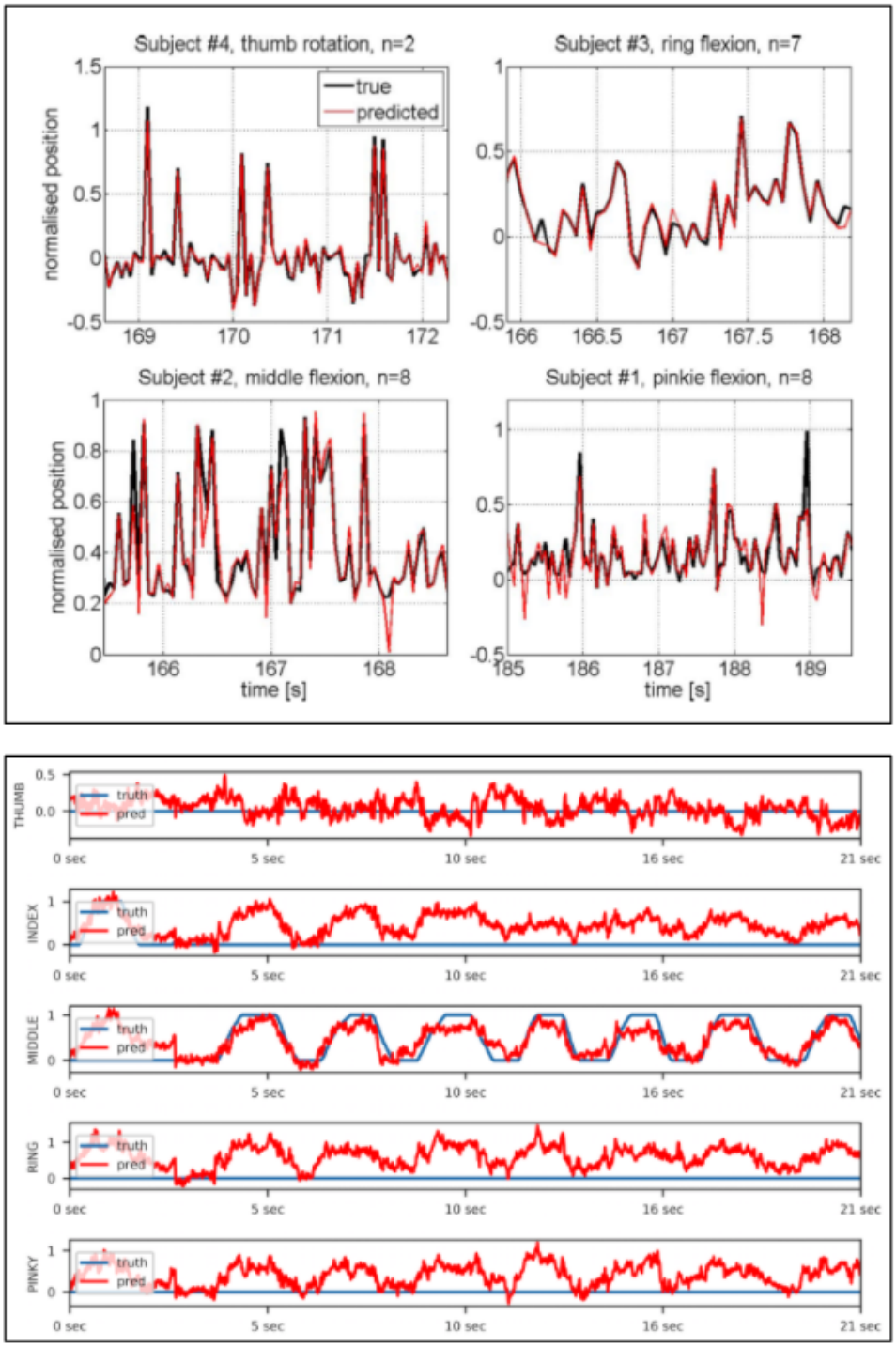


Figure 6.12: **Top:** A plot taken from Castelinni et al. [14] showing true and predicted normalized for different finger flexions. For example, the middle flexion has reasonably high regression accuracy, but the values for the other fingers at the same instance in time are not shown. **Bottom:** Linear Regression from our models trained on ultrasound echoes. We are able to get qualitatively similar performance on the middle finger, but also show a complete picture of other finger models failing on the same input sample.

fingers when a target finger is being predicted are not shown (Figure 6.12 on page 63). This is significant, as we were able to obtain qualitatively similar plots to Castelinni et al. using our linear regression baseline on ultrasound echoes, but also show how the simultaneous predictions from other fingers contain false activations in our baseline approach.

6.3 Experiment 2: Thresholded Classification

6.3.1 Task Description

In this section, we use the same models trained in experiment 1, but threshold the results into discrete classes. We do this for two reasons, firstly, to demonstrate the viability of "switching" modes from a regression model into a classifier, without need to train a new system just for classification. Secondly, to provide a means of comparing to previous work. However, there are limitations to this comparison, since previous work ignore arm orientations completely. Moreover, many of the previous approaches using SEUS transducers explicitly noted their tests as "preliminary" in nature. In the some cases, the subject and samples per subjects sizes are much smaller than the ones employed in this thesis. Thus, comparison of metrics should be regarded as a qualitative assessment.

6.3.2 Dataset Division

Our original dataset consists of real valued flexion vectors as ground truth. To threshold these into discrete classes, we first remove all mixed samples containing simultaneous flexions so the remaining dataset consists only if individual finger flexions. We then set an upper threshold of 0.7 and a lower threshold of 0.3. These values are arbitrary and not derived from data. We only do so because calculating metrics such as accuracy, precision and recall for comparison require an explicit threshold to be set. Thus we define samples below the lower threshold to be "open" hand samples while samples with a flexion above 0.7 are considered flexions. Since there are no mixed finger samples, we can simply extract the index of max in the flexion vector to determine what finger was flexed. We omit all samples

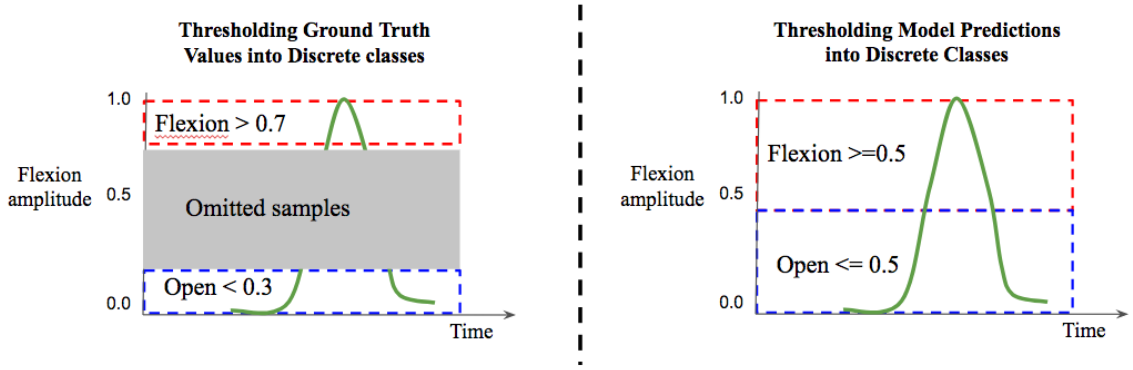


Figure 6.13: **Left:** in the ground truth data, samples with flexion ¿0.7 are considered positive finger flexions for that finger. Samples with flexions ¡0.3 are considered open hand and all other samples in the middle dead-band are discarded. **Right:** During inference, the model will output relatively noisy outputs (i.e. the model will never regress a pure flexion vector of [0,0,0,0,0]). We thus choose a threshold of 0.5 as the cutoff between open hand vs a positive flexion for the finger.

in the "dead-band" region. This divides the continuous dataset into 6 discrete classes: open, thumb, index, middle, ring and pinky. The classes were all balanced. Metrics are evaluated on the unseen test set defined in Experiment 1.

6.3.3 Thresholding model output

Recall the regression output is noisy. For example, the model will never output a true flexion vector of [0,0,0,0,0] for open hand. Thus we set a threshold of 0.5; values above are considered flexion while values below are considered open hand. In addition to the thresholding of ground truth labels, we acknowledge this thresholding of model output is arbitrary, and only do so because in order to compare to previous work.

A much better assessment of classifier performance is the Receiver Operating Characteristic (ROC). The advantage of using ROC curves lies in how it eliminates the notion of thresholding. The true positive rate and false positive are calculated for all thresholds and the ROC curve is plotted for each individual finger. The Area Under Curve (AUC) can then be calculated from the ROC curve, with 1.0 indicating perfect classification for all thresholds. Unfortunately, previous work do not provide the ROC-AUC metric for direct

Finger Classes	Precision	Recall	f1-score	samples
Open	0.33	0.93	0.48	152390
Thumb	0.96	0.73	0.83	215472
Index	0.93	0.70	0.80	213898
Middle	0.82	0.73	0.77	213003
Ring	0.87	0.51	0.64	216959
Pinky	0.89	0.61	0.72	211827
Average	**0.83**	**0.69**	**0.72**	1223549

Table 6.5: Discretized Holdout Test metrics. Samples denote the number of data points over which the metrics were calculated (not the number used to train the classifier). Metrics averaged over 50 total folds across all 10 users. The difference in precision and recall point towards non-optimal thresholds

comparison in this manner.

6.3.4 Results

Table 6.5 on page 66 shows the Precision, Recall and F1-score of the model as a classifier thresholded at 0.5. We report an average F1-score of 0.72, Precision of 0.83 and Recall of 0.69.

Figure 6.14 on page 67 shows the confusion matrix from our model with the output regression thresholded at 0.5. The class accuracies are reported from a testing set of $\sim$1,200,000 samples. We report a high classification accuracy of 93% for the open hand class, approximately 70% for the thumb, index and middle fingers. The ring finger has the worst classification accuracy at 51% and little finger at 61%.

Figure 6.15 on page 68 shows the ROC curve based on the model regression output with no arbitrary thresholding. When calculating ROC for a multi-dimensional vector such as the flexion vectors encountered in this work, the ROC can be calculated per class, and then averaged into a combined value through a micro or macro mean [1]. To calculate the ROC per class, we extract only regressions for that finger (e.g. extract the entire of column no.2 in the (n,5) prediction matrix for index predictions) and treat the continuous output as the

[1]Micro vs. Macro average ROC. `http://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html`

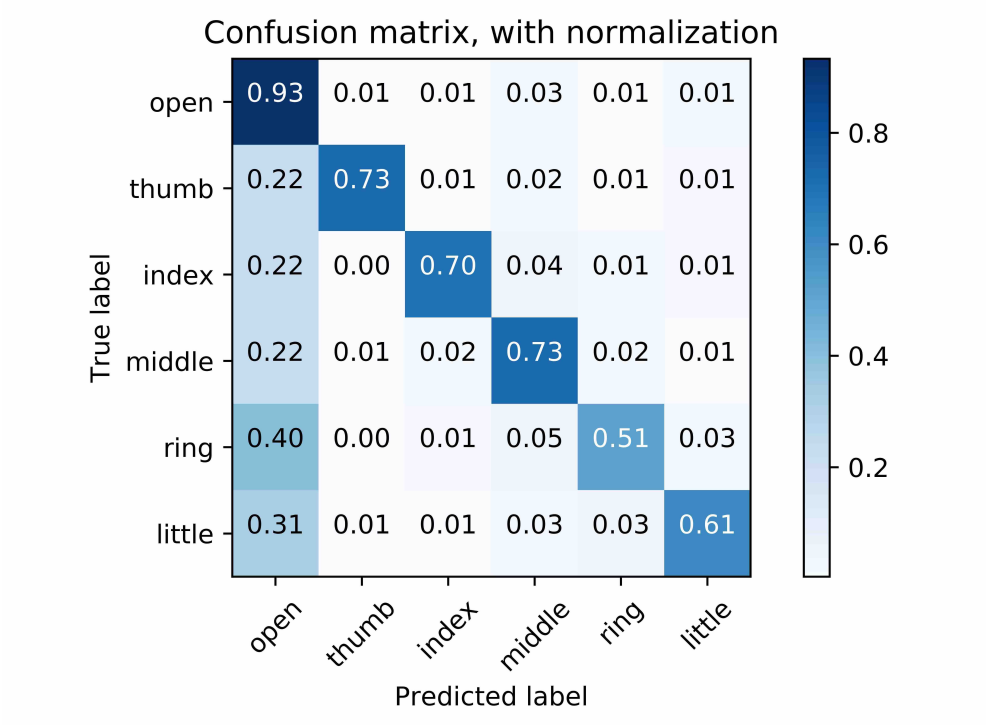


Figure 6.14: Confusion Matrix with normalization. Class accuracies are computed from a total number of ∼1,200,000 test set samples.

confidence level of a binary classifier. The true positive rate and true negative rate can then be calculated for a range of thresholds, and the ROC curve for that finger can be plotted. We argue this is a more apt assessment of performance a classifier, as it does not depend on the arbitrary threshold.

The inset of 6.15 shows the Area Under Curve (AUC) score for the ROC plot per finger, as well as the AUC for the micro and macro average curve. We report high ROC-AUC values, with each class scoring at or exceeding 0.90 (AUC has a maximum of 1.0). The micro and macro average both are at 0.93.

6.3.5 Discussion

Classification metrics

The F-1 Score of 0.72 on Table 6.5 on page 66 provides evidence that supports the use of the model as a classifier over a range of arm orientations, a development over previous work.

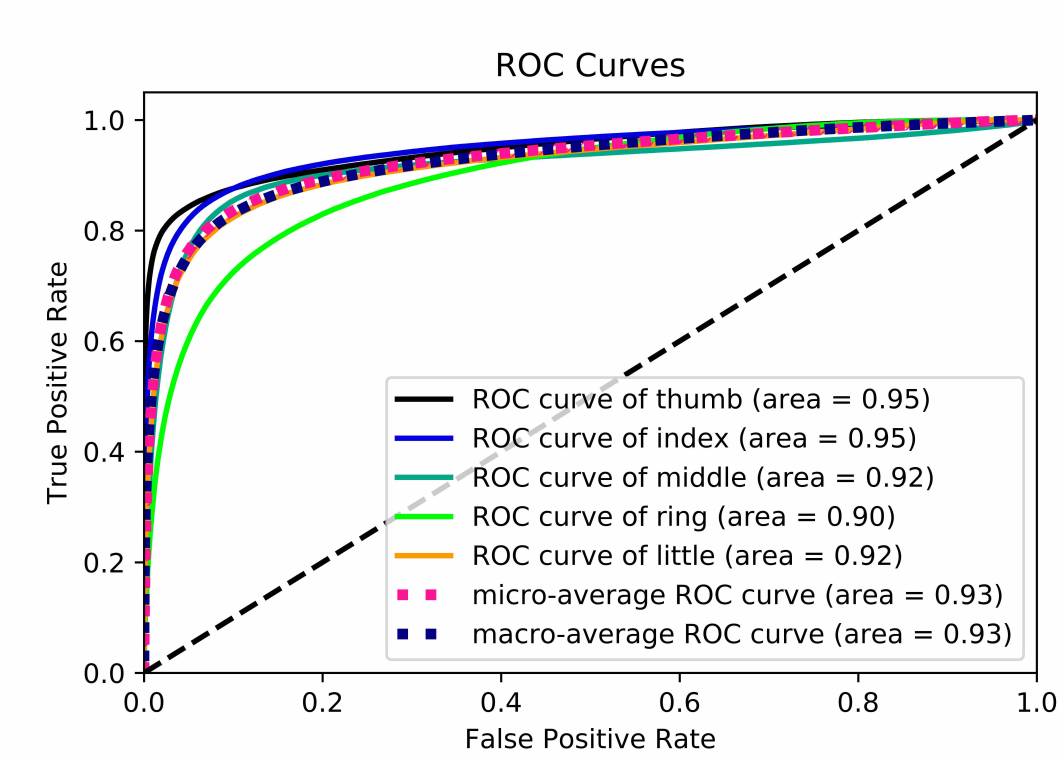


Figure 6.15: ROC curve per finger. Inset contains AUC for each finger in addition to micro and macro averages across all fingers

These metrics should be regarded as baselines for future work to improve on through the hypotheses motivated in experiment 1 and later in Chapter 7: Conclusion and Future work. However, there are discrepancies between the average precision and recall, 0.83 and 0.69 respectively. This trend is apparent across all the classes and points towards 0.5 being a non-optimal threshold for the classifier.

The confusion matrix show in 6.14 on page 67 show promising performance as a classifier, as evidenced by the strong diagonal. However there is still a strong vertical corresponding to misclassification of finger flexions as open hand. This may be due to the model being biased to open hand, although we believe it more likely an artefact of thresholding. Notice how thumb number of thumb, index and middle fingers misclassified as open hand samples is roughly equal, $\sim$22%. However, the ring and pinky are misclassified at a much higher rate ($\sim$40% and $\sim$31% respectively), suggesting it is relatively more likely for a ring or pinky to be misclassified as an open hand sample than other fingers. From visual inspection of the ultrasound echoes discussed earlier, the ring and pinky fingers have characteristic echo patterns very similar to open hand. Thus we would expect classification accuracy of these classes to be extra sensitive to the choice of thresholding.

For this reason, we use the ROC and AUC metric on our model regression in order to assess classification accuracy. The 6.15 on page 68 as well as AUC metrics in the figure inset support the use of our model as a classifier. Individual AUC metrics per finger are all greater than 0.9, with particularly good performance achieved by the thumb and index fingers. These metrics also provide evidence that the ring finger does perform worse compared to fingers like the thumb and index fingers, which could be due to physiological, hardware and machine learning limitations that should be explored in a future study.

Comparison with previous work

Despite significant differences in a US image and US echo, overall classification accuracies shown in Table 6.6 on page 70 provide a rough and qualitative estimate on the trade off when

Work	Acc.	No. Sensors	Classes	Subjs.	Samps./Sujb.
SEUS-CNN (**with rot.*)	0.71	5	6	10	∼120,000
Li et al. (*prelim.*) (2016) [15]	0.95	4	6	3	∼6000
Ortenzi et al. (2015) [9]	0.92	image	10	3	∼10,000
Sikdar et al. *(prelim.)* (2014) [10]	0.98	image	5	10	∼70

Table 6.6: Comparison of our approach with previous work using images or SEUS transducers for a classification task. Note that previous work do not account for arm rotation and orientation. The Samples per Subject are estimated from the methodology and frame rates described in the corresponding publication.

performing on classification on images versus echoes. In its current iteration, ultrasound echoes do not perform as well as ultrasound images, but the overall accuracy of 0.71 is a promising starting point. The high degree of class separation enabled the amputated musician to reliably control a prosthetic arm to play a melodic sequence on the piano.

Table 6.6 on page 70, comparing our classification accuracy with previous work using SEUS transducers and ultrasound images. These comparison are made using the non-optimal thresholding of 0.5 on the regression output from the model. In the case of Sikdar et al. [10], we can perhaps ascertain the high accuracy to be due to a much smaller Sample per Subject size. The same argument could be argued with Li et al. [15], which only had 3 subjects in their experimentation. Both works cite their methodology as being preliminary in nature. Ortenzi et al. [9] were able to get over 90% using an LDA classifier trained on HOG features. This provides evidence that there is a performance drop-off when moving from ultrasound images to raw ultrasound echoes. However, our model's weaker performance may also be due to the effects of accounting for arm orientation.

We propose two future studies. The first involves filtering our current dataset to a narrow subset of arm orientations, simulating a situation where the arm is "static" and enabling more grounded comparison with previous work. This nonetheless has limitations, since some previous work do not explicit document the arm position the data was collected at. Alternatively, a second experiment could re-implement the methodologies in previous works on the dataset collected in this thesis. This will provide stronger grounds for comparing

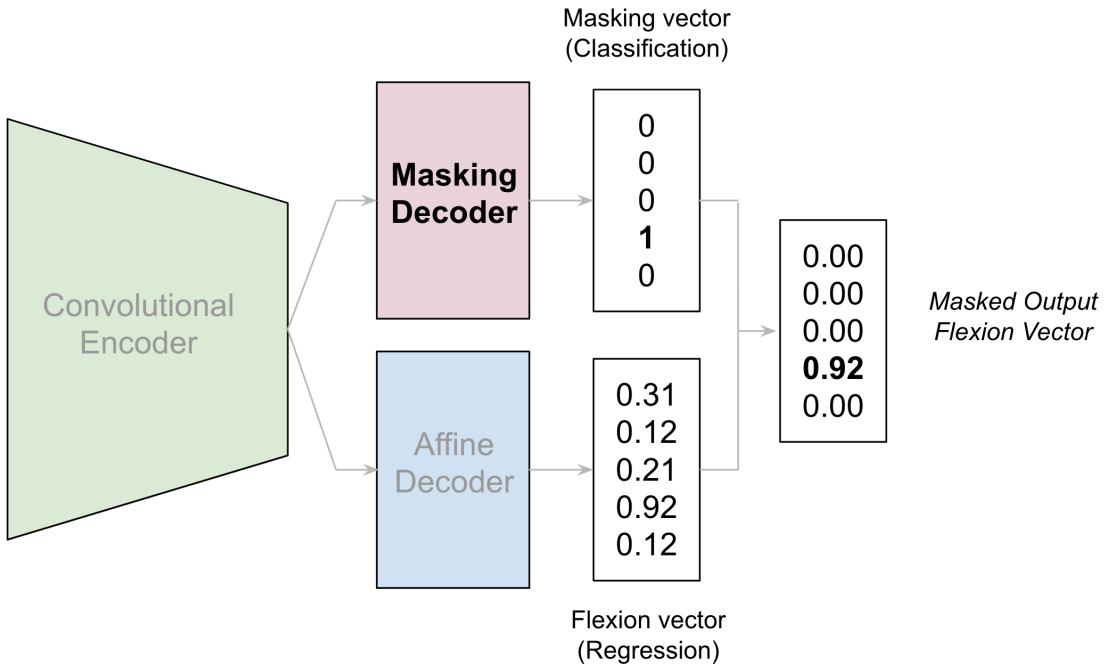


Figure 6.16: Structure of an alternative network combining both regression and classification into a single task. In addition to the affine decoder, a second masking decoder is tasked with predicting which other fingers to "zero-out" in order to reduce the error when comparing to the ground truth data. The model implicitly learns to do both classification and regression in this approach.

previous approaches with our SEUS-CNN and be beneficial for the ultrasound research community at large. The size and scope of data collected in this thesis would be an excellent follow up publication to these works, since our dataset is order of magnitudes larger than those collected in previous experiments and is applicable in both the ultrasound image and echo domain.

6.4 Obtaining Regression and Classification Simultaneously

The results and discussions from the separate regression and classification experiments in this chapter point towards a smarter implementation that unifies regression and classification into a single model for this dataset. We previously noted in the regression experiment how the model is wrongly penalized for predicting related flexions of other fingers that happen naturally when the labeled finger is flexed. We could reduce the error the model is making

on our dataset by masking the output flexion vector to reflect the dataset by "zero-ing" out all other fingers not being flexed. This can be achieved by training a second masking decoder shown in Figure 6.16 on page 71 that is tasked with predicting what finger is being moved and what fingers should be zeroed. The masking decoder would output a softmax classification over the five fingers. The final output is a masked flexion vector that only contains the regressed value at the desired finger, much like the ground truth annotations.

The advantage of this approach is two-fold. Firstly, the model will be able to achieve better scores across all metrics, including lower MAE, MSE and RMSE and higher R^2 values. Secondly, the model has implicitly learned the task of classification, since the masking vector corresponds to the finger being flexed! This approach avoids the notion of training a separate regression model and then thresholding the flexion output to turn the model into a classification model. Previous work have shown how multi-task learning can yield better performance in the individual sub-tasks [27]. The dual-task nature of this model may even yield better results for the individual tasks of regression and classification.

CHAPTER 7
CONCLUSION AND FUTURE WORK

7.1 Conclusion

In this work, we motivate ultrasound as a viable alternative HMI device interface. We collected a large and comprehensive dataset of ultrasound images and echoes over 10 users, and trained a multi-modal convolutional neural network that could regress finger flexions with promising accuracy across different arm orientations. The same model also shows strong performance as a classifier for discrete hand gestures. Using ultrasound as an HMI provides advantages such as not being limited by line of sight (vision HMI devices), independence from sweat interference or muscle fatigue (sEMG) and freedom from unnatural finger obstructions (data gloves). The ultrasound hardware presented in this thesis can be miniaturized into a small band and worn similar to a watch. At the time of writing, our research group has already begun capitalizing on new CMUT (Capacitative Micromachined Ultrasonic Transducers) transducers that are smaller than the ones used in this thesis and most importantly, flexible. This technology could eventually be integrated into the band of wearable consumer technology like smart-watches and fitness trackers. The user would be able to interface with computers, TV's and other devices using this always-wearable and accurate finger tracking sensor as a general purpose HMI.

7.2 Future Work and Recommendations

This thesis is the first implementation of an ultrasound HMI that combines single element ultrasound transducers, non-stationary operation and recent advancements in machine learning. Over the course of this thesis, the author has motivated several future studies based on unanswered questions from the results presented in this work.

We propose and detail the two most relevant experiments and studies that should immediately follow this work.

7.2.1 Multi-Task Learning

We designed a network structure that enforces filter invariance across sensors. However, a new model is trained from scratch for each user. Ideally, we would want to transfer the features learned from User A to help in the task of learning features for User D. One possible approach is to train the model designed in this paper using data from 9 of the users, and then transfer the learned weights to the 10th user. This would be a transfer learning task, where we would hope features learned from 9 users will enable a model to output accurate predictions on the 10th user with a smaller number of samples or with a faster training time.

A more exciting and promising approach is using multi-task learning, where each user is considered a "task". This structure of training would enforce the model to learn features invariant across users. Due to time and computation constraints, the author was unable to complete comprehensive testing on this approach with conclusive evidence. The experimentation approach is documented and motivated here in detail.

Task Description

Given a set of 10 related tasks T, learn a function $f(x, \theta)$ that minimizes the global minimum of these tasks combined. This is shown in 7.1 on page 75. Here, there are 3 inputs to the network, corresponding to the ultrasound data from each user. These are then passed through a shared convolutional encoder, which branches out into 3 outputs. When the network is trained, data from user A is passed through the network a loss A is computed, then the same is repeated for user B and user C. The 3 losses are summed and back propagated through the network. This structure of training forces the network to generalize across users by learning features that are useful to all users, or useful to all tasks T. Each user also has a unique set of decoding layers that can be concatenated with accelerometer input.

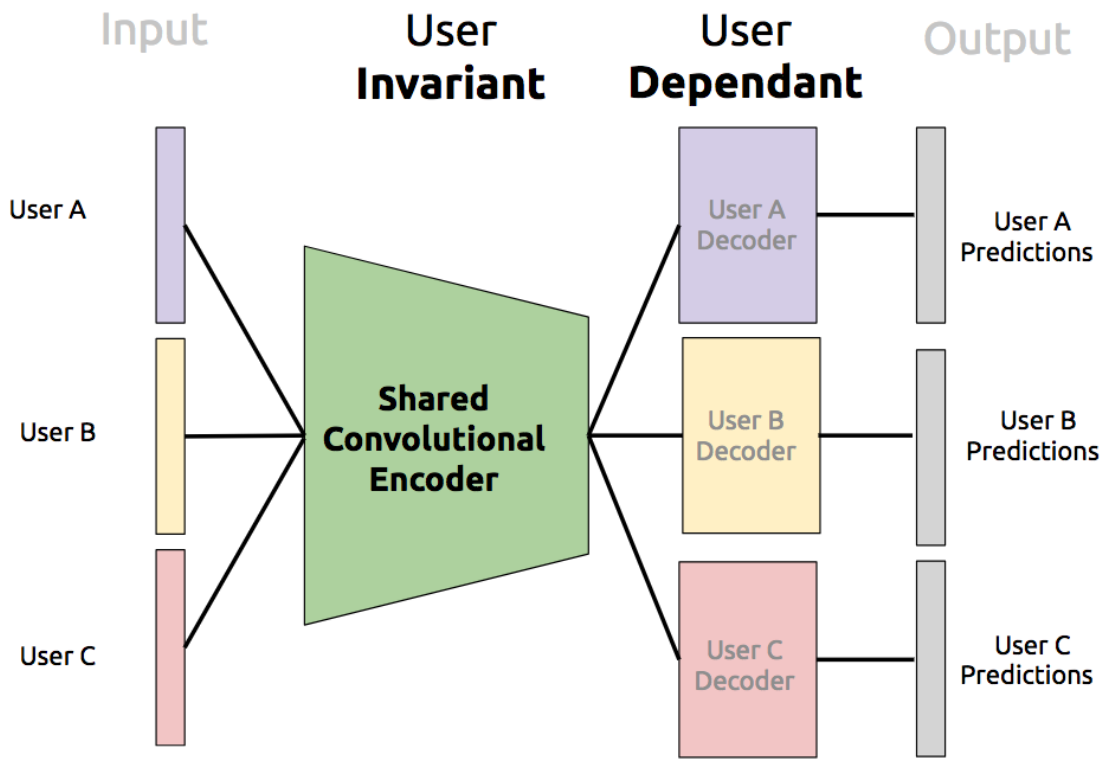


Figure 7.1: Structure of the network with a shared layer, but multiple inputs and multiple outputs. The losses from the 3 users are summed at the end and back propagated through the network)

Motivation

This particular approach seems particularly apt to the task of ultrasound echoes because of the following reasons:

- Every user is different: Each user has a set of unique ultrasound echo patterns relating to their unique physiology. The TSNE plot in Figure 6.1 on page 43 shows how samples from a user will group strongly with other samples from the same user, with clean separation from another users. There is strong intra-class grouping and strong inter-class separation. In other words, each task T is non-identical.

- There are some similarities across users: Despite a set of unique ultrasound echo patterns per user, the task of mapping these waveforms to flexion vectors per user involves the same process of learning: identify peak location and morphology, then map changes in these features to the flexion vectors. In other words, each unique task T has a set of related characteristics.

We would use the weights learned from this convolutional encoder on a new user D, keeping the weights fixed and only training the decoding layer specific to the user. We would hope that training on the new user D either converges faster or requires a smaller amount of data when the architecture is held constant. To show this, the experiment must train a network with random initialization for user D (as done in this thesis) and compare the performance with a network initialized with weights from the multi-task experiment. The experiment should use increasing percentages of data, to ensure the model trained with random weights was not already performing at a good level.

It is also possible to enforce greater generalization by randomizing which 3 users will be chosen per epoch of training. This way, the model doesn't know which group of users it will get and thus must learn features that will be relevant to all users. Readers may wonder why not just train the model jointly on 10 users. Previous research suggests this number may be too high for effective learning. Multi-task learning typically ties together 3 - 5 tasks

[37]. In this experiment, the reader must take care to also swap the user-dependent decoders to correspond to with each user's input data (i.e. don't have User E's data back propagated through User A's dependent decoder)

This approach will likely require the model's capacity to be increased. We suggest these changes to model architecture for future exploration:

- **Atrous/Dilated convolutions:** In addition to increasing the number of layers, the dilation rate should increase deeper in the model. As discussed before, this enables the model to retain a wider receptive view deeper int the network.

- **Residual Connections:** ResNets [38] have shown remarkable success in image classification by enabling the model to use the activations of a previous layer in addition to activations from the current layer. This identity function prevent gradients and information from being "lost" as the network depth increases. We believe this may be highly applicable to our task, since the two main components are peak location and peak morphology. The activations from a previous layer may be more likely to preserve overall locations of peaks, whereas activations from the current layer may contain more information on peak morphology.

- **Recurrent CNN's:** As discussed earlier, our approach enforces a 1:1 mapping between input and output. However, stacking multiple frames to make a prediction may yield better performance. On top of this, a recurrent model that makes use of previous states and predictions when making a current prediction, may be beneficial. Given a sequence of flexion vectors, it is possible to predict what the next flexion vector will be since it is physiologically impossible to jump from a thumb flexion to an index flexion at the frame rates collected in this work. Recurrent Convolutional network structures have been previously studied [39, 40] and can be easily applied to this work by replacing the 2D filters with the 1D filters employed in this work

7.2.2 Musical Applications

There are many applications of finger flexion tracking in domains such as prosthetics, virtual reality and music. The author is most excited about applications in music performance and composition. Fast, accurate, expressive and lightweight controllers are essential to new musical interfaces and controllers, such as those found in the New Interfaces for Musical Expression (NIME) and International Computer Music Conference (ICMC) communities. Many of these wearable controllers have also expanded into the commercial market and popular music scene, being used by well-known musicians such as Imogen Heap [1].

At the time of writing, the ultrasound band developed in this thesis requires an expensive, large and difficult to move ultrasound machine housing the specialized Analog to Digital Converters (ADC) and pulsing circuitry. This has limited the potential of easily using this system in a musical or performance setting, both as a standalone device and as part of an ensemble. Nonetheless, our group is currently developing a portable version of this setup using FPGA chips and embedded devices such as the NVidia Jetson TX2 [2]. We are also actively developing a version of this research that could run on a smartphone and interface with a custom-made ultrasound hardware. This next portable iteration will be an excellent opportunity to explore the device in a musical context.

A musical study should evaluate the controller as an alternative controller to popular devices such as the Leap Motion and Data Gloves. Some of these studies are proposed below:

Musical Perceptual Study

A perceptual study could involve users moving their fingers and mapping this movement directly to pitch. As a control, this study would be conducted with minimal sonic parameters. Users would simply be controlling glissando of a sinusoidal oscillator, followed by a

[1] Mimu Gloves. https://mimugloves.com/

[2] Jetson TX2 Module. https://www.nvidia.com/en-us/autonomous-machines/embedded-systems-dev-kits-modules/

questionnaire that asks participants to rate on a scale of 1 - 5 on metrics such as:

- How “jumpy” or “smooth” was the glissando pitch they playd
- How “late” or “on time” were the changes in pitch in relation to their finger movements

These questions seek to quantify the perceptual latency and perceptual continuity of our ultrasound system in a musical setting. The same experiment could be repeated and compared with other sensors such as a Leap Motion.

Evaluation using a pre-designed musical system

This is a more musically sophisticated Three forms of controllers: a vision-based device like the Leap Motion, a Data Glove and our ultrasound system, are mapped to an interactive music system designed to generate musical sounds such as pitches or soundscapes. Both musicians and non-musicians should be recruited for this study. The user is asked to perform a series of musical tasks such as playing a melody with ascending pitches for 5 seconds, holding that pitch for 3 seconds and then playing a set of descending pitches. The user repeats these musical tasks using each system and ranks them through questionnaire questions like:

- How easy was it to acquire and hold the desired pitch using the system?
- How responsive was the system to your finger movements?
- How was the size, weight and ease of setup?
- What were the most difficult aspects using the interface?

Evaluation through the design of a new interactive music system

In this study, musicians are asked to compose and perform a short piece using the three aforementioned controllers. The musician is given an overview of the main section of

the pieces, such as points of high note density or low energy, but is free to compose the music and interactive system in any way they choose. The study will be conducted in two time spans, a short 1 - 2 hour session and a longer period across several days. The aim of the short experiment is to evaluate the first usage of our system in comparison to other systems. Does the musician use it to control aspects of pitch? Or do musicians tend to use our ultrasound system to control parameters related to timbre or soundscaping? The aim of the longer experiment is to give enough time for musicians to become acclimated with the ultrasound system, perhaps finding novel ways of using the system beyond its original purpose of detecting finger flexions. The author motivates some of these creative uses in the next section.

Performance techniques unique to the Ultrasound system

Artistic communities often find new and unconventional ways to use new sensor technologies. A great example is the use of the GameTrak Golf Game Controller , a peripheral with two strings that attach to the user's hands and tracks their golf swing in-game. The device was instead used as an accurate tracking mechanism for hand location for use in controlling sonic parameters in a laptop orchestra [41].

No commercial device currently uses ultrasound as the sensing mechanism. The ultrasound advantage lies in how the fingers are left unhindered (unlike a data glove) and the user is free from sensor field of view (unlike the Leap Motion and Kinect). This enables the user to simply strap the device on like a watch or fitness tracker and simply use their hands and fingers normally. The user can grasp and lift objects, perform high dexterity tasks like cooking or crafting and play musical instruments, all unhindered. This opens up unprecedented interactive opportunities unique to our system:

- A pianist could perform a piece that leverages the data from their tendon activations as the song is played. Whereas sEMG systems require strong and well defined muscle activations to accurately detect muscle changes, our system relies on changes in

Figure 7.2: Unique example usages of our ultrasound system as an artistic or musical interface

muscle and tendon morphology. Thus, the small, intricate and highly controlled finger flexions from a pianist or another instrumentalist can be detected by our system. Subtle changes in morphology related to arm rotations as the user moves across the instrument, could all be registered and mapped to sound-scaping parameters. A data glove would interfere with the pianist's playing while a Leap Motion would be unable to track fingers across the piano keyboard length.

- Since the band is located on the wrist, user's can interact with fluids or sensor-occluding objects. For example, consider a piece that uses our ultrasound band with a clay sculptor. A data glove is impossible to use, since it directly hinders the sculptor's craft and the electronics will likely be damaged by the clay. As the sculptor reaches into a mound of wet clay, their fingers become occluded during this entire process of manipulation and will be lost by vision-based sensors. sEMG is out of the question, since it would require the sculptor to unnaturally flex and move their muscles to the point of interfering with their regular craft. The ultrasound band would still detect finger flexions in all these above circumstances with little to no hindrance to the sculptor.

- Simple activities like typing on a computer keyboard or opening a bottle, could become new and exciting interactive sonified pieces. A performance could involve musicians interacting with items that produce very little sound but require careful fin-

ger control like typing, and use our ultrasound system to meaningfully sonify muscle activations associated with these activities.

- The ultrasound band opens up interactions where the physical accordances of an object, and its affect on the human body, become the main form of interactive control. For example, given a fixed mapping of finger flexions to sound parameters, a box containing different kinds of fluids with different viscosities will cause different muscle activations. Flexing a group of fingers in air versus flexing the same group of fingers in cookie dough could yield interesting shades in sound character.

- Mapping ultrasound echoes to finger force, as opposed to flexion, would enable strenuous activities to be registered by our system. For example, lifting a heavy weight will cause stronger tendon activations than lifting a light feather, even though both actions require similar finger flexions. This could enable users to incorporate object weight, density and morphology into musical performance. Instead of the user physically manipulating the properties of an object (e.g. a box containing accelerometers and gryoscopes detecting orientation) which are in turn mapped to sonic parameters, it is the properties of the object (the weight, texture and shape) that changes the user's physical manipulation of it, which in turn would be detected by our ultrasound band.

CHAPTER 8

CONTRIBUTORS

All contributors involved in this ongoing project and thesis are collected and recognized in the follow section:

Undergraduate students: Kipp Morris, Sahaj Bhatt, Wai Man Si, Sereym Baek, Heather Song, Yoel Molinas, Chaewon Minh, Naoto Abe

Graduate students: Zachary Kondak MSMT'18, Keshav Bimbraw MSMT'19

Post Doctoral Fellows: Bernie Shih, Coskun Tekes, Chris Fink

Principal Investigators: Professors Gil Weinberg, Levent Degerteken and Minoru Shinohara

Computing resources: Lenovo Inc. and NVidia Corp.

Appendices

APPENDIX A

TRANSDUCER TECHNICAL SPECIFICATIONS

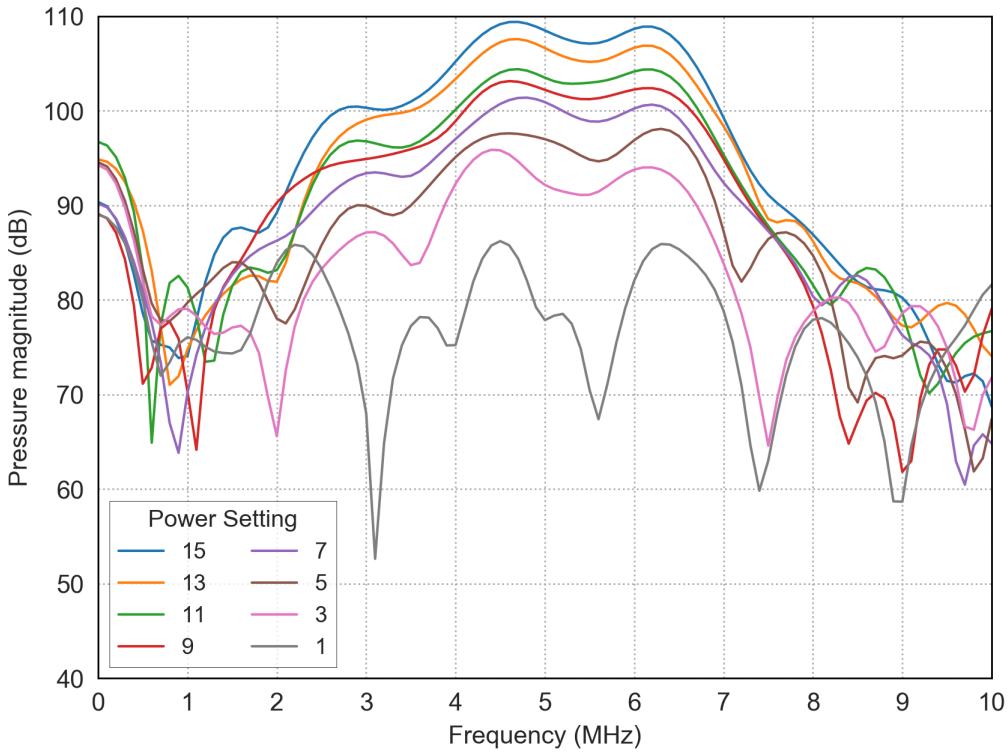


Figure A.1: Frequency response of ultrasound transducers. Figure provided by Bernie Shih

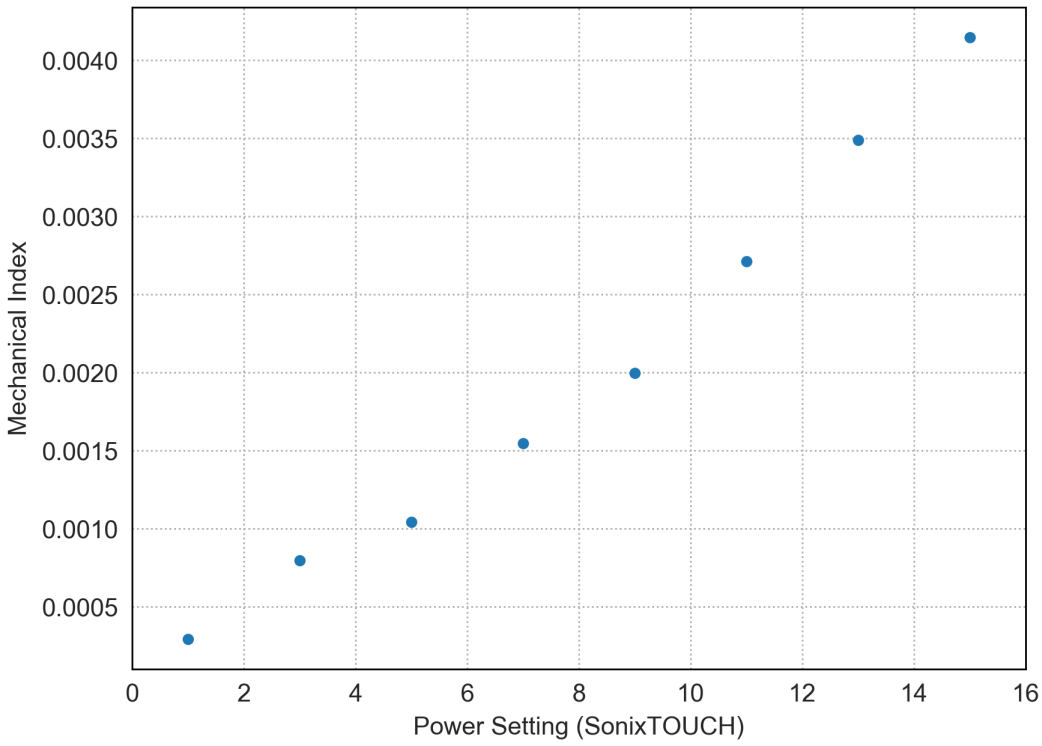


Figure A.2: Mechanical Index of Single element ultrasound transducers. Figure provided by Bernie Shih

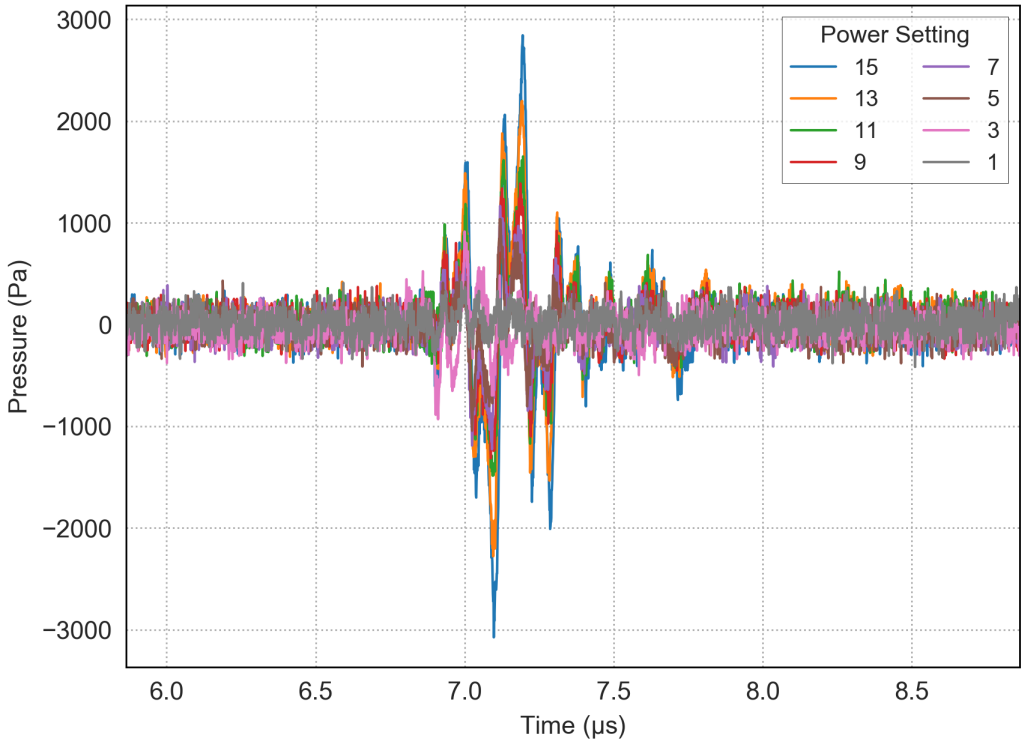


Figure A.3: Pulse emitted by the ultrasound transducer. Figure provided by Bernie Shih

APPENDIX B

EXPERIMENT 1 RESULTS

User/Finger	Thumb	Index	Middle	Ring	Pinky	**All**	Samples
cm-foldav	0.49	0.504	0.395	0.31	0.334	0.406	(332284, 5)
hh-foldav	0.852	0.737	0.722	0.494	0.645	0.69	(302638, 5)
hs-foldav	0.832	0.849	0.838	0.707	0.709	0.787	(317920, 5)
km-foldav	0.896	0.758	0.67	0.712	0.789	0.765	(296743, 5)
na-foldav	0.597	0.681	0.566	0.486	0.593	0.585	(284470, 5)
rs-foldav	0.685	0.602	0.731	0.52	0.643	0.636	(319227, 5)
sb-foldav	0.699	0.634	0.782	0.458	0.647	0.644	(330481, 5)
ss-foldav	0.782	0.61	0.555	0.374	0.517	0.567	(342210, 5)
ym-foldav	0.625	0.739	0.762	0.599	0.658	0.677	(276735, 5)
zk-foldav	0.715	0.698	0.244	0.528	0.593	0.555	(336256, 5)

Table B.1: Test Set R^2 Full Results

User/Finger	Thumb	Index	Middle	Ring	Pinky	**All**	Samples
cm-foldav	0.679	0.673	0.621	0.562	0.578	0.623	(332284, 5)
hh-foldav	0.891	0.836	0.826	0.698	0.794	0.809	(302638, 5)
hs-foldav	0.906	0.917	0.922	0.849	0.843	0.887	(317920, 5)
km-foldav	0.932	0.862	0.826	0.838	0.876	0.867	(296743, 5)
na-foldav	0.753	0.825	0.771	0.712	0.763	0.765	(284470, 5)
rs-foldav	0.793	0.75	0.822	0.711	0.771	0.769	(319227, 5)
sb-foldav	0.828	0.769	0.859	0.675	0.784	0.783	(330481, 5)
ss-foldav	0.89	0.789	0.77	0.644	0.737	0.766	(342210, 5)
ym-foldav	0.799	0.871	0.876	0.785	0.819	0.83	(276735, 5)
zk-foldav	0.852	0.851	0.672	0.745	0.787	0.781	(336256, 5)

Table B.2: Test Set Pearson Correlation Full Results

User/Finger	Thumb	Index	Middle	Ring	Pinky	**All**	Samples
cm-foldav	0.095	0.115	0.14	0.163	0.134	0.129	(332284, 5)
hh-foldav	0.043	0.087	0.093	0.159	0.09	0.094	(302638, 5)
hs-foldav	0.058	0.053	0.06	0.086	0.073	0.066	(317920, 5)
km-foldav	0.034	0.068	0.078	0.069	0.05	0.06	(296743, 5)
na-foldav	0.073	0.075	0.092	0.116	0.096	0.091	(284470, 5)
rs-foldav	0.068	0.108	0.077	0.111	0.078	0.088	(319227, 5)
sb-foldav	0.08	0.089	0.062	0.127	0.087	0.089	(330481, 5)
ss-foldav	0.061	0.081	0.096	0.134	0.107	0.096	(342210, 5)
ym-foldav	0.108	0.086	0.085	0.136	0.102	0.104	(276735, 5)
zk-foldav	0.075	0.09	0.164	0.148	0.122	0.12	(336256, 5)

Table B.3: Test Set Mean Absolute Error Full Results

User/Finger	Thumb	Index	Middle	Ring	Pinky	**All**	Samples
cm-foldav	0.045	0.046	0.054	0.06	0.052	0.051	(332284, 5)
hh-foldav	0.02	0.035	0.041	0.068	0.037	0.04	(302638, 5)
hs-foldav	0.018	0.015	0.014	0.026	0.024	0.019	(317920, 5)
km-foldav	0.011	0.023	0.029	0.027	0.02	0.022	(296743, 5)
na-foldav	0.036	0.026	0.039	0.045	0.035	0.036	(284470, 5)
rs-foldav	0.028	0.038	0.029	0.043	0.029	0.034	(319227, 5)
sb-foldav	0.029	0.033	0.021	0.045	0.029	0.031	(330481, 5)
ss-foldav	0.019	0.035	0.038	0.052	0.04	0.037	(342210, 5)
ym-foldav	0.036	0.026	0.028	0.05	0.035	0.035	(276735, 5)
zk-foldav	0.034	0.03	0.086	0.056	0.045	0.05	(336256, 5)

Table B.4: Test Set Mean Squared Error Full Results

User/Finger	Thumb	Index	Middle	Ring	Pinky	**All**	Samples
cm-foldav	0.471	0.482	0.312	0.246	0.298	0.362	(177005, 5)
hh-foldav	0.848	0.713	0.737	0.469	0.617	0.677	(161677, 5)
hs-foldav	0.821	0.831	0.819	0.707	0.675	0.771	(169936, 5)
km-foldav	0.881	0.753	0.683	0.704	0.763	0.757	(157612, 5)
na-foldav	0.606	0.67	0.624	0.468	0.591	0.592	(151630, 5)
rs-foldav	0.702	0.564	0.723	0.513	0.63	0.626	(170219, 5)
sb-foldav	0.698	0.618	0.743	0.446	0.599	0.621	(175887, 5)
ss-foldav	0.738	0.6	0.541	0.356	0.517	0.55	(182497, 5)
ym-foldav	0.612	0.706	0.765	0.56	0.64	0.657	(147277, 5)
zk-foldav	0.689	0.631	0.244	0.454	0.578	0.519	(178904, 5)

Table B.5: Validation Set R^2 Full Results

User/Finger	Thumb	Index	Middle	Ring	Pinky	**All**	Samples
cm-foldav	0.692	0.695	0.573	0.51	0.549	0.604	(177005, 5)
hh-foldav	0.922	0.845	0.864	0.686	0.786	0.821	(161677, 5)
hs-foldav	0.906	0.911	0.905	0.842	0.823	0.877	(169936, 5)
km-foldav	0.939	0.868	0.837	0.839	0.875	0.872	(157612, 5)
na-foldav	0.78	0.819	0.805	0.699	0.773	0.775	(151630, 5)
rs-foldav	0.84	0.753	0.851	0.717	0.795	0.791	(170219, 5)
sb-foldav	0.858	0.787	0.862	0.668	0.774	0.79	(175887, 5)
ss-foldav	0.859	0.781	0.736	0.606	0.719	0.74	(182497, 5)
ym-foldav	0.785	0.848	0.875	0.749	0.802	0.812	(147277, 5)
zk-foldav	0.835	0.802	0.661	0.679	0.76	0.747	(178904, 5)

Table B.6: Validation Set Pearson Correlation Full Results

User/Finger	Thumb	Index	Middle	Ring	Pinky	**All**	Samples
cm-foldav	0.096	0.119	0.154	0.167	0.137	0.135	(177005, 5)
hh-foldav	0.044	0.091	0.089	0.159	0.092	0.095	(161677, 5)
hs-foldav	0.058	0.056	0.062	0.087	0.079	0.068	(169936, 5)
km-foldav	0.037	0.07	0.078	0.068	0.05	0.061	(157612, 5)
na-foldav	0.075	0.08	0.086	0.115	0.094	0.09	(151630, 5)
rs-foldav	0.067	0.11	0.08	0.117	0.082	0.091	(170219, 5)
sb-foldav	0.079	0.09	0.067	0.12	0.088	0.089	(175887, 5)
ss-foldav	0.063	0.085	0.102	0.136	0.108	0.099	(182497, 5)
ym-foldav	0.113	0.093	0.083	0.136	0.1	0.105	(147277, 5)
zk-foldav	0.078	0.098	0.163	0.16	0.122	0.124	(178904, 5)

Table B.7: Validation Set Mean Absolute Error Full Results

User/Finger	Thumb	Index	Middle	Ring	Pinky	**All**	Samples
cm-foldav	0.042	0.045	0.061	0.066	0.054	0.053	(177005, 5)
hh-foldav	0.015	0.033	0.033	0.069	0.038	0.037	(161677, 5)
hs-foldav	0.018	0.016	0.017	0.027	0.026	0.021	(169936, 5)
km-foldav	0.01	0.022	0.027	0.026	0.021	0.021	(157612, 5)
na-foldav	0.032	0.027	0.031	0.046	0.033	0.034	(151630, 5)
rs-foldav	0.023	0.038	0.025	0.042	0.026	0.031	(170219, 5)
sb-foldav	0.023	0.03	0.02	0.044	0.03	0.029	(175887, 5)
ss-foldav	0.023	0.036	0.042	0.057	0.041	0.04	(182497, 5)
ym-foldav	0.039	0.03	0.028	0.054	0.037	0.038	(147277, 5)
zk-foldav	0.038	0.04	0.09	0.068	0.049	0.057	(178904, 5)

Table B.8: Validation Set Mean Squared Error Full Results

REFERENCES

[1] D. Sierra González and C. Castellini, "A realistic implementation of ultrasound imaging as a human-machine interface for upper-limb amputees," *Frontiers in neurorobotics*, vol. 7, p. 17, 2013.

[2] B. Rahmatullah, A. T. Papageorghiou, and J. A. Noble, "Image analysis using machine learning: Anatomical landmarks detection in fetal ultrasound images," in *Computer Software and Applications Conference (COMPSAC), 2012 IEEE 36th Annual*, IEEE, 2012, pp. 354–355.

[3] K. T. Dussik, "Über die möglichkeit, hochfrequente mechanische schwingungen als diagnostisches hilfsmittel zu verwerten," *Zeitschrift für die gesamte Neurologie und Psychiatrie*, vol. 174, no. 1, pp. 153–168, 1942.

[4] M. Fink, *Imaging of complex media with acoustic and seismic waves*. Springer Science & Business Media, 2002, vol. 82.

[5] T. M. Jørgensen, A. Tycho, M. Mogensen, P. Bjerring, and G. B. Jemec, "Machine-learning classification of non-melanoma skin cancers from image features obtained by optical coherence tomography," *Skin Research and Technology*, vol. 14, no. 3, pp. 364–369, 2008.

[6] J. Zhang, K.-K. Ma, M.-H. Er, and V. Chong, "Tumor segmentation from magnetic resonance imaging by learning via one-class support vector machine," in *International Workshop on Advanced Image Technology (IWAIT'04)*, 2004, pp. 207–211.

[7] J. Shi, Q. Chang, and Y.-P. Zheng, "Feasibility of controlling prosthetic hand using sonomyography signal in real time: Preliminary study," *Journal of rehabilitation research and development*, vol. 47, no. 2, p. 87, 2010.

[8] C. Castellini, K. Hertkorn, M. Sagardia, D. S. González, and M. Nowak, "A virtual piano-playing environment for rehabilitation based upon ultrasound imaging," in *Biomedical Robotics and Biomechatronics (2014 5th IEEE RAS & EMBS International Conference on*, IEEE, 2014, pp. 548–554.

[9] V. Ortenzi, S. Tarantino, C. Castellini, and C. Cipriani, "Ultrasound imaging for hand prosthesis control: A comparative study of features and classification methods," in *Rehabilitation Robotics (ICORR), 2015 IEEE International Conference on*, IEEE, 2015, pp. 1–6.

[10] S. Sikdar, H. Rangwala, E. B. Eastlake, I. A. Hunt, A. J. Nelson, J. Devanathan, A. Shin, and J. J. Pancrazio, "Novel method for predicting dexterous individual finger movements by imaging muscle activity using a wearable ultrasonic system," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 1, pp. 69–76, 2014.

[11] Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE multimedia*, vol. 19, no. 2, pp. 4–10, 2012.

[12] B. Carse, B. Meadows, R. Bowers, and P. Rowe, "Affordable clinical gait analysis: An assessment of the marker tracking accuracy of a new low-cost optical 3d motion analysis system," *Physiotherapy*, vol. 99, no. 4, pp. 347–351, 2013.

[13] N. Jiang, S. Dosen, K.-R. Muller, and D. Farina, "Myoelectric control of artificial limbsis there a need to change focus?[in the spotlight]," *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 152–150, 2012.

[14] C. Castellini, G. Passig, and E. Zarka, "Using ultrasound images of the forearm to predict finger positions," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 20, no. 6, pp. 788–797, 2012.

[15] Y. Li, K. He, X. Sun, and H. Liu, "Human-machine interface based on multi-channel single-element ultrasound transducers: A preliminary study," in *E-Health Networking, Applications and Services (Healthcom), 2016 IEEE 18th International Conference on*, IEEE, 2016, pp. 1–6.

[16] N. Hettiarachchi, Z. Ju, and H. Liu, "A new wearable ultrasound muscle activity sensing system for dexterous prosthetic control," in *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*, IEEE, 2015, pp. 1415–1420.

[17] A. Hafiane, P. Vieyres, and A. Delbos, "Deep learning with spatiotemporal consistency for nerve segmentation in ultrasound images," *ArXiv preprint arXiv:1706.05870*, 2017.

[18] T. L. Szabo, *Diagnostic ultrasound imaging: Inside out*. Academic Press, 2004.

[19] U. Food and D. Administration. (2017). Ultrasound imaging, (visited on 04/12/2017).

[20] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, "Analysis of the accuracy and robustness of the leap motion controller," *Sensors*, vol. 13, no. 5, pp. 6380–6393, 2013.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[22] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[23] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning.," in *AAAI*, vol. 4, 2017, p. 12.

[24] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[25] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, IEEE, 2009, pp. 248–255.

[27] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 689–696.

[28] L. Kaiser, A. N. Gomez, N. Shazeer, A. Vaswani, N. Parmar, L. Jones, and J. Uszkoreit, "One model to learn them all," *ArXiv preprint arXiv:1706.05137*, 2017.

[29] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh, and D. Batra, "Visual dialog," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2017.

[30] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for lvcsr using rectified linear units and dropout," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, IEEE, 2013, pp. 8609–8613.

[31] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[32] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *ArXiv preprint arXiv:1609.03499*, 2016.

[33] Y. Aytar, C. Vondrick, and A. Torralba, "Soundnet: Learning sound representations from unlabeled video," in *Advances in Neural Information Processing Systems*, 2016, pp. 892–900.

[34] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International Conference on Machine Learning*, 2016, pp. 173–182.

[35] P. Casale, O. Pujol, and P. Radeva, "Human activity recognition from accelerometer data using a wearable device," in *Iberian Conference on Pattern Recognition and Image Analysis*, Springer, 2011, pp. 289–296.

[36] C. Castellini, E. Gruppioni, A. Davalli, and G. Sandini, "Fine detection of grasp force and posture by amputees via surface electromyography," *Journal of Physiology-Paris*, vol. 103, no. 3-5, pp. 255–262, 2009.

[37] S. Ruder, "An overview of multi-task learning in deep neural networks," *ArXiv preprint arXiv:1706.05098*, 2017.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[39] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, IEEE, 2015, pp. 4580–4584.

[40] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*, 2015, pp. 802–810.

[41] B. Ruviaro, "From schaeffer to* lorks: An expanded definition of musical instrument in the context of laptop orchestras," in *Proceedings of the 1st Symposium on Laptop Ensembles & Orchestras*, 2012, pp. 23–26.